

CatBoost-based Data Science Software for Lung Cancer Risk Estimation: Bayesian Validation and p@k Evaluation for Imbalanced Data

Manoel Vilela Machado^a, Helena Oliveira^{a,b}, Oscar de Almeida Machado^{a,b,c}, Felipe Alves Neto Verri^a

^a*Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, São José dos Campos, 12228-900, São Paulo, Brazil*

^b*Universidade Federal de São Paulo, Avenida Cesare Mansueto Giulio Lattes, 1201 - Jardim Santa Ines I, São José dos Campos, 12247-014, São Paulo, Brazil*

^c*Academia Militar das Agulhas Negras, Rod Presidente Dutra Km 312, Paraíso, Resende, 27533-050, Rio de Janeiro, Brazil*

Abstract

This paper aims to describe a software development initiative where integrating Machine Learning tools enhances project capabilities. Adhering to 10 key principles for data science projects ensures the software's efficiency and reliability. Bayesian validation guided model selection, ensuring optimal performance, thereby improving accuracy and streamlining decision-making processes. These methodologies enable effective navigation of the complexities inherent in data-driven projects, ensuring robust outcomes aligned with project goals. The integration of Machine Learning into software development fosters innovation, harnesses data-driven insights, and significantly enhances overall project success and user experience.

Keywords: Machine Learning, Data Science, Bayesian Validation, Lung Cancer, Detection Risk

1. Introduction

According to data from the Pan American Health Organization PAHO (2024), lung cancer is one of the leading causes of mortality worldwide, accounting for approximately 18% of cancer-related deaths globally. Early diagnosis of this type of cancer is crucial for increasing the chances of cure, as survival rates are closely linked to the stage of the disease at the time

of diagnosis. According to World Health Organization WHO (2024), some studies indicate that when detected in the early stages, cancer is more likely to respond to treatment and can result in a greater probability of survival with less morbidity, as well as less expensive treatment.

The importance of early diagnosis is underscored by the high incidence and mortality associated with lung cancer. Also according to the WHO (2024), in 2020, approximately 2.2 million new cases of lung cancer were recorded, along with 1.8 million deaths resulting from the disease. These alarming statistics highlight the urgent need to improve early detection methods, which can potentially save countless lives.

The development of technological tools for the prediction and estimation of lung cancer risk plays a fundamental role in this scenario. The use of advanced software based on machine learning algorithms and artificial intelligence has shown promise in analyzing large volumes of clinical data and identifying subtle patterns that may indicate an increased risk of developing the disease.

Recent studies have demonstrated that such tools can achieve superior accuracy compared to traditional risk assessment methods, providing a more personalized and effective approach to patient screening. Quanyang et al. (2024) mention that with the advancements and implementation of cutting-edge technologies, such as neural networks and deep learning algorithms, the potential for AI applications in lung cancer screening has been continuously explored.

In the literature, it is common to find several studies that used Machine Learning models in the lung cancer prediction. Chandra et al. (2023) used machine learning to develop a 3-year lung cancer risk prediction model with large real-world data in a mostly younger population. Richter and Khoshgoftaar (2018) made a review of the most Machine Learning Models used for the cancer prediction. They concluded that many studies utilize statistical survival analysis techniques. Those that do not use survival analysis build predictive models using machine learning techniques such as Decision Trees, Neural Networks, and Support Vector Machines.

Based on the above premises, it can be observed that the creation and validation of lung cancer risk estimation software represent a significant advancement in the fight against this devastating disease. This work describes the development, implementation, and evaluation of new software designed to improve the accuracy of early lung cancer diagnosis.

This software is addressed to the general practitioners in order to referring

a patient with a high risk of lung cancer to an oncologist is an essential measure. This professional can perform a more accurate and early diagnosis of the disease, which significantly increases the patient’s chances of survival. Additionally, an early diagnosis allows for the adoption of more effective treatment strategies, improving the patient’s quality of life.

The software was developed in four modules: frontend, backend, dataset, and model search. The frontend module basically is the user interface. The backend module is the server-side code. The dataset is the the data that is used to train the model. Finally, the model search is the code that searches for the best model.

In Figure 1, we present the overall software architecture pertaining to the solution’s modules. On the application side, a REST API serves the backend web application and integrates the model binary produced by the model search algorithm, which identifies the optimal model for the data. The database constitutes the persistence layer, storing patient data and predictions based on a front-end survey, which will be detailed in subsequent sections. Finally, the deployment infrastructure connects all components and exposes the service on the internet. The circular flow represents a potential workflow within the data science cycle.

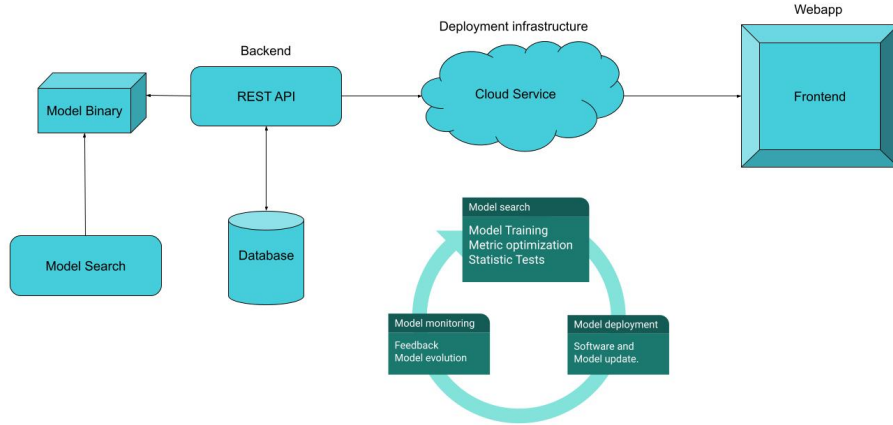


Figure 1: Design of the Lung Cancer Risk Estimation Software. Source: Authors.

2. Data Collection and Preprocessing

A search was conducted on the internet on dataset websites such as UCI, Kaggle, and sites affiliated with cancer-fighting foundations, such as the American Cancer Foundation, for example.

The objective of the software is to enhance the detection of lung cancer risk during the initial ambulatory medical assessments, with general practitioners being the primary users of this software. Therefore, all datasets containing mostly images of exams or other data that can only be extracted after an imaging exam were discarded, focusing solely on symptomatic data.

In this regard, three datasets were selected according to the following methodology described.

2.1. Premises

Health applications dealing with diseases often encounter challenges related to *class imbalanced data*. This typically means there is an abundance of data available for patients with the disease but insufficient data for healthy controls. Moreover, these data sources often exhibit varying attributes. Integrating these sources into a unified dataset that closely mirrors real-world distributions is critical for applications aimed at predicting risks within the general population.

While some studies focus on overall survival rates among diagnosed individuals or those undergoing treatment, or individuals presenting initial indicators such as flags found in lab results or medical suspicion of cancer, this research specifically aims to estimate unknown risks for random individuals based on survey data. Subsequent iterations may incorporate additional variables such as air quality, demographics, and healthcare inputs.

2.2. Data Strategy

Datasets 1 and 2 (abbreviated as ds1 and ds2) served as the training data, combined to enhance sample size and variability. To address the inherent bias toward high disease risk, a validation dataset (ds3) regarding COVID-19 ¹ was introduced for having patients with comparable features. Despite ds3 not containing the cancer target variable, its known disease incidence within the population provided a basis for evaluating model performance.

The datasets and its features are described in appendix, table 5.

¹COVID-19: a respiratory viral illness, which caused a pandemic and resulted in ample availability of related data

	ds1	ds2	ds3
Number of rows	309	1000	316800
Number of columns	16	26	27
Features characteristics	Mostly binary	Mostly range 1-10	Mostly binary
Target variable	LUNG_CANCER	LEVEL	-
Target distribution	NO: 39 YES: 270	High: 365 Medium: 332 Low: 303	-

Table 1: Dataset Characteristics. Source: Authors

2.3. Feature Engineering

Lacking a specific knowledge of the domain of expertise alongside a broad understanding of epistemology, ontology, and logic, may result in incorrectly perceiving phenomena, leading to inaccurate data collection and ultimately flawed conclusions, Verri (2024, p. 37). Domain knowledge and scaling were used for data integration.

Practical applications may require additional techniques. Consulting health professionals while integrating data for health applications is advisable. When dealing with continuous variables from various sources, such as lab test results, it is essential to ensure consistent scales and measurement methods.

When applying feature transformation, two primary options were considered. The first option and chosen option involved converting features into binary format. This approach simplifies categorical data into a more straightforward format suitable for various machine learning algorithms, including CatBoost, despite involving information loss. The second option would take the features to a continuous form, representing a symptom level. This method preserves the nuanced information inherent in continuous variables, which can be crucial for models sensitive to magnitude and variation.

In the implementation discussed here, one-hot encoding was select to transform categorical variables, also well-suited for CatBoost algorithm, which was chosen for its robust handling of categorical data and its ability to manage complex interactions between variables. This approach facilitates clearer interpretation and effective utilization of categorical data within the machine learning model.

Matching

As highlighted by Silva et al. (2011), **ontology** plays a crucial role. Despite having domain knowledge, we may encounter synonymous terms in features naming, which could be integrated. Conversely, some features with identical wording may represent different concepts. Examples of matching implemented in this study dataset integration:

- TIREDNESS \equiv FATIGUE
- DIFFICULTY_IN_BREATHING \equiv SHORTNESS_OF_BREATH
- SORE_THROAT \equiv SWALLOWING_DIFFICULTY

Feature generation

The study in Bitencourt et al. (2022) explores the creation of artificial variables through the use of coefficients that represent key measures of students' academic performances. This approach utilizes machine learning techniques to tackle the issue of student accomplishments across diverse schools and programs. Due to varying numbers and subjects of courses taken by students, direct data joins result in substantial missing data. To mitigate this issue, the study establishes connections between objects by generating composite indicator features for each student in the dataset. Unlike constructing an index, the method presented here involves grouping symptoms to develop new features customized for the specific context.

The strategic feature generation not only addresses data incompleteness but also enhances the predictive capabilities of machine learning models, by reducing redundancy and enriching the dataset with more informative features, these variables can contribute to more robust and accurate predictions.

To enhance the predictive power of the model, artificial variables were introduced through a process of calculation and integration. Specifically, a new variable named **RESPIRATORY_SYMPTOMS** was created by aggregating or applying weighted values to existing respiratory-related variables across multiple datasets (ds1, ds2, ds3). Each dataset's specific formulas were tailored to capture the unique characteristics and relationships among these variables, ensuring that the derived artificial variables accurately reflect the underlying patterns and complexities present in the data.

Features are typically designed by domain experts who possess in-depth knowledge of the data properties, including its nature and intended applications. This expertise often results in what are known as hand-crafted features. While they may require manual effort initially, these features leverage

human insight, making them particularly straightforward to interpret, (Dong and Liu, 2018, p. 56) .

Target Variable Transformation

The target variable was redefined as **lung cancer risk**, to avoid the misunderstanding that the classifier will be answering if the person has the disease or not, even though the feature, in ds1, originally stands for "does the person have been diagnosed with lung cancer?" . We progressed with the assumption that people with lung cancer diagnostic has high lung cancer risk. In ds2, we find the levels low-medium-high, table 1, we defined, for this implementation, low and medium as non-positive for lung cancer risk. The target variable in ds3 remained missing.

3. Model Search

The model search began before the completion of dataset integration. Initially, a data exploratory analysis (DEA) of the selected datasets was conducted. A Proof of Concept (POC) was used, employing the logistic regression model, as it was defined that the program output would be a flag indicating to the physician that the patient had a high risk of developing lung cancer.

After the partial completion of dataset integration, it was concluded that the logistic regression model would not be suitable, as there were categorical variables in the integrated dataset. Therefore, it was suggested to implement the CatBoost Classifier model, as this model handles categorical variables.

The CatBoost Classifier model was subsequently adjusted to CatBoost Regression. This change provided greater flexibility to the model, once a threshold (τ) was defined to transform it into a classification model, where the predicted risk $f_C(x)$ of the patient having cancer would have a value of 1 (high risk) if the risk predicted by the model $f_R(x)$ was equal to or greater than the threshold. Otherwise, the predicted risk would assume a value of 0 (not high risk). In mathematical terms, we have:

$$f_c(x) = \begin{cases} 1, & \text{if } f_R(x) \geq \tau \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In this model, the precision at k (P@k) metric was implemented. This metric is used to evaluate the accuracy of a recommendation or classification system based on the top k recommended items. It measures the proportion

of relevant items among the top k recommendations, particularly valuable in scenarios where the order of recommendation matters, such as search results or personalized recommendations.

The $P@k$ metric was chosen to validate the model because the training dataset contains many missing labels, which makes traditional metrics such as accuracy, F1 score, and precision impractical to use. $P@k$ evaluates the top k recommendations by analyzing the labeled data that corresponds to those top recommendations.

Cross-validation technique was also employed to ensure greater robustness of the model. The cross-validation technique divides the dataset into r folds randomly, with each fold having the same size. Each fold is then used as a test set once and as a training set $r - 1$ times. Initially, folds 2 through r are used as the training set and fold 1 as the test set. Subsequently, folds 1, 3, \dots , r are used as the training set, and fold 2 as the test set, and so forth. For a better understanding of this technique, please refer to figure 2:

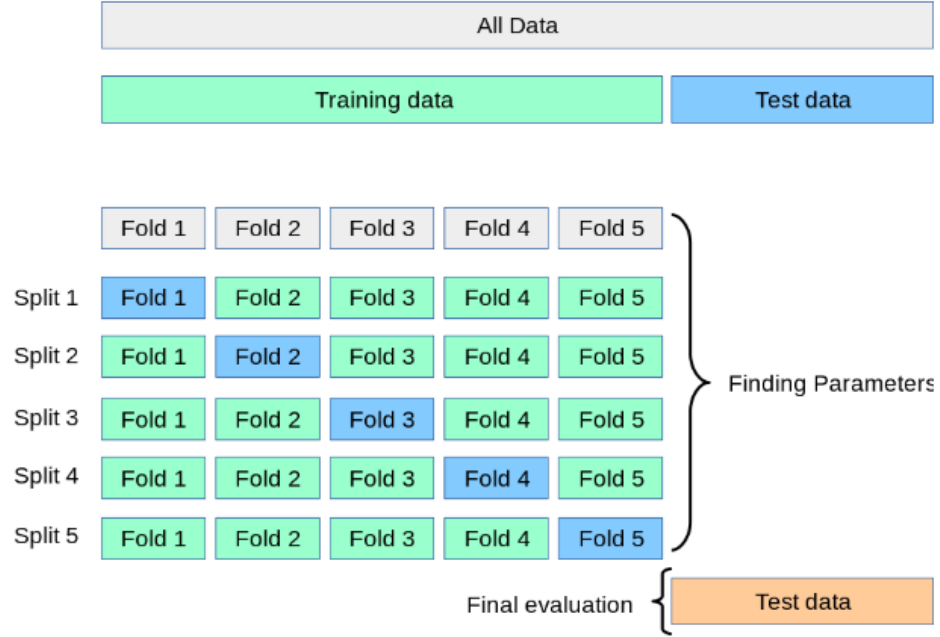


Figure 2: Cross-validation with 5-folds scheme. Source: Scikit-learn documentation. Available at https://scikit-learn.org/stable/modules/cross_validation.html. Accessed on June 10, 2024.

Concerning the model evaluation, a Bayesian validation was conducted.

This technique serves as an alternative to traditional methods of model evaluation, which employ the conventional Null Hypothesis Significance Testing (NHST). There are some studies that criticize the use of NHST. Benavoli et al. (2017) is one such study, presenting a Bayesian validation technique where the likelihood function is a multivariate normal noise with zero mean and a given covariance matrix. The prior distribution is a Normal-Gamma. By combining both distributions, a t-Student distribution is obtained.

In this work, we will not delve into the study of Bayesian validation. For more details, see Benavoli et al. (2017) and Corani and Benavoli (2015). To perform such technique, we use a 10-fold with 10-times cross-validation as in Benavoli et al. (2017). The library "verri-evc", developed by Prof Verri of Aeronautics Institute of Technology was used, whose code is available at link in the footnote ².

To finalize the model selection and simplify the dataset used, Bayesian validation was conducted again to choose the features that could be dropped from the model. Initially, the feature importance function was used to identify which features had the greatest impact on the model. Subsequently, a correlation matrix was created to assess how the features were interrelated. Features with strong correlation (close to 1) were examined and proposed to be dropped.

At this point, three models were proposed, based on the model containing all the features. For the second model, 14 features were dropped, and for the third model, these 14 features plus another 4 were dropped. Additionally, a fourth model was developed as a control, where only the most important features were dropped.

All code used in the model search was documented using GitHub. More details about the code can be viewed at link ³.

The results of search model will be presented in the section 6.

4. Application

As mentioned earlier, the application has been architecturally modularized into frontend and backend components, facilitating seamless integration with model search functionality and its versioned artifacts in a Data Version Control (DVC) repository, as delineated in the preceding sections. In regard

²<https://github.com/verri/evc>


³<https://github.com/cancer-estimator>

of the technologies, for frontend we used mainly HTMX for frontend-backend integration through HTML tags and Bootstrap for styling, the backend uses FastAPI as HTTP Server and SQLite as database. The deployment is made as a Docker container through cloudflare tunnels to expose to internet using a on-premise server hosted by one of the member of our team.

4.1. Frontend

Regarding the frontend development of our application, we have implemented a survey profile application focused on symptomatic features to emulate a potential electronic health record (EHR) system. This system comprises two primary interfaces: the main page, which features a search bar for querying and registering new patients, and the profile page, where patient details and symptoms pertinent to lung cancer risk assessment can be recorded.

Search patients

 Patient

Patient	Room	Lung Cancer High Risk
Lorena Lana	20-B2	True
João	20-B	False
Verri	20-B	True

Add new patient

Figure 3: Main page with search bar. Source: Authors.

In Figure 3, we present the main interface of our frontend application. This interface includes a search bar that allows users to search for specific patient names or to add new patient records. By selecting any row from

the search results, the application navigates to the Patient Profile page, as depicted in Figure 4.

Patient Profile

[← Back](#)

Name:

Lorena Lana

Sex:

Female

Age:

42

Patient Room:

20-B2

☒ Has medical diagnosis of lung cancer

☒ Hospitalized

[Save](#) [Add extra symptoms for lung cancer risk estimation](#)

Figure 4: Patient profile page with basic information. Source: Authors.

In Figure 4, users can input fundamental patient information, such as name, sex, age, and additional details. Some of these details serve as features in the risk estimation model, while others, such as patient room and hospitalization status, are included for administrative purposes. The flag "has medical diagnosis of lung cancer" is optional and is designed to facilitate future feedback regarding true-positive and false-positive outcomes when the system is deployed in a production environment over an extended period.

Upon activation of the "Add extra symptoms for lung cancer risk estimation" button, a set of 22 symptom-based questions is presented to facilitate accurate lung cancer risk estimation. These questions were meticulously designed utilizing model search, feature selection, and feature importance

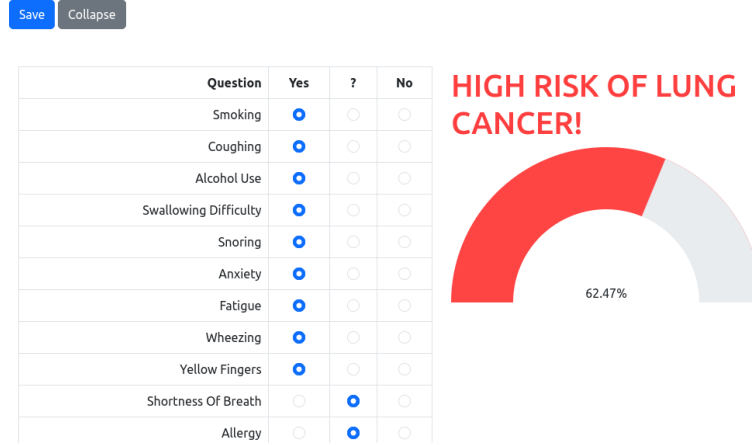


Figure 5: Patient profile symptoms survey and cancer risk estimation. Source: Authors.

methodologies, as elucidated in prior sections. Following the completion and saving of a sample of responses, an alert flag for cancer risk may be triggered if the regression model’s threshold of 0.6 is exceeded, as depicted in Figure 5.

4.2. Backend

For the backend part of the application, we developed two main parts, where we can receive the patient cancer risk score and its flag using a REST API, as well integrated to emit a HTML page properly formatted to user-read and interaction. In that case, we can get a lung cancer estimation score using only the REST API (machine-readable) or using the web application in a more user-friendly way.

4.3. Development Workflow

Our application includes a CI/CD pipeline powered by Github Actions, enabling us to release new versions of both the model and application directly. We manage a dedicated repository for model search, storing artifacts like trained models in a shared DVC repository. Additionally, our application repository combines both the backend and frontend components. The following diagram at Figure 6 summarizes the software engineering interaction steps.

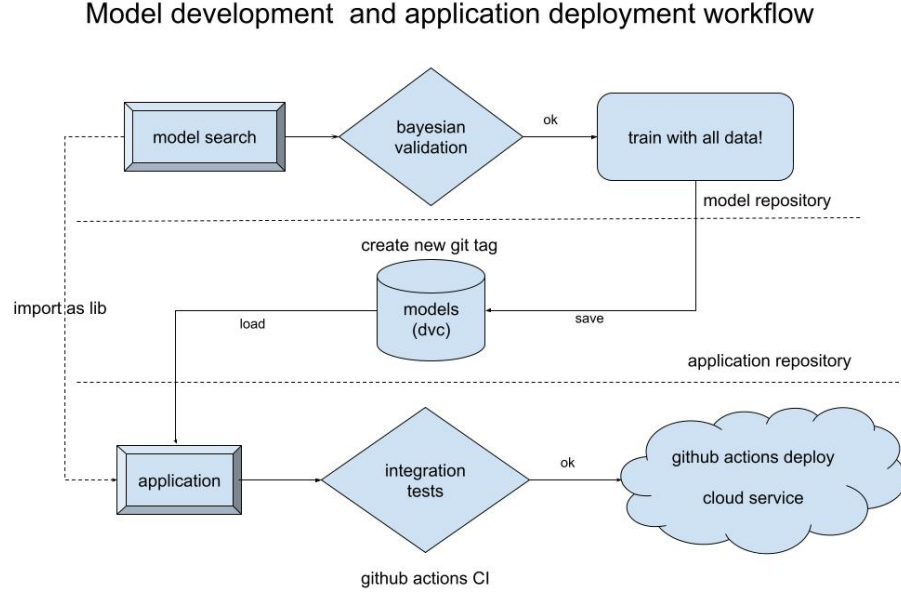


Figure 6: Software workflow deployment between application and model search. Source: Authors.

5. Theory

The main theoretical reference guiding this work, although not the only one, is Vapnik’s Statistical Learning Theory. The deep dive into the mathematical aspects behind the most commonly used models currently is not within the scope of this study, but for more details about Statistical Learning Theory, see Vapnik (1999). Since the presented problem is related to classification, for the purposes of this study, we employ tree-based models such as Decision Tree, Random Forest, and Catboost.

However, a data science project involves merging typical machine learning tools as described with software development tools. Thus, it is emphasized that certain principles related to software development can and should be observed in the development of data science software. Additionally, some fundamental principles of machine learning must be taken into account. To assist in the software development process, we follow 10 basic principles.

According to Verri (2024), these principles are: Software Modularity, Ver-

sion Control, Continuous Integration and Continuous Deployment (CI/CD), Report as deliverables, Setup quantitative goals, Measure exactly what you want, Report Model Stability and Performance Variance, Avoid data-science-specific terminology in the user interface, Monitoring model performance in production and Use the appropriate backend.

In our approach, software modularity is foundational, dividing our solution into Dataset, Model Search, Frontend, and Backend components. This segmentation not only enhances maintainability and scalability but also promotes component reusability. Each module can be developed and updated independently, facilitating efficient management and evolution of the entire system without disrupting its core functionality.

Version control is pivotal across our project's facets: code, datasets, and task tracking. Git manages code versions and task assignments, while DVC (Data Version Control) handles dataset versioning. This comprehensive approach ensures that changes are tracked systematically, promoting transparency and facilitating collaborative development. Effective versioning supports rigorous monitoring of project evolution, ensuring that updates are managed seamlessly across different project phases.

Our development methodology integrates Continuous Integration and Continuous Deployment (CI/CD), fostering agility and reliability. By automating testing and deployment processes, CI/CD enables frequent updates while maintaining a stable and deployable software state. This iterative approach not only enhances software quality but also accelerates feedback cycles, enabling rapid iteration and responsiveness to evolving project requirements.

Client engagement is central to our project success. We maintain regular communication through weekly progress meetings, where we report on project milestones and solicit feedback. This proactive approach ensures alignment with client expectations and allows for timely adjustments, fostering a collaborative and client-focused development environment.

Setup quantitative goals is crucial in this project because since the goal of model accuracy was reached, we can prioritize other tasks. Programmers often worry about endlessly improving the accuracy of the model and end up neglecting other important tasks.

The principle "measure exactly what you want" can be translated into the appropriate choice of metric for the model. In the developed software, the metric $p@k$ was chosen, as mentioned earlier, which is more suitable for evaluating the model's performance.

The following principle, "Report Model Stability and Performance Variance," implies a better understanding of the model's limitations and characteristics. As mentioned earlier, some assumptions were used in the development of the software, which were accepted by the client as valid, with an understanding of the limitations arising from this acceptance.

Since the software was designed to be used by a user who does not belong to the field of data science, technical terms specific to this area were not used in the user interface.

To meet the ninth principle, "Monitoring model performance in production," a button was introduced where the user can provide feedback on whether the individual developed the disease or not. This will enable continuous feedback for the model.

Due to the fact that user interface updates do not require constant updates and can be done in batches, the decision was made to use REST API as mentioned previously in section 4.2.

Adhering closely to these principles was crucial in developing the software within a short timeframe and with a high level of confidence.

6. Results

To validate the model using the P@k metric, the crucial question is to define the optimal value for k and also the ideal threshold for the metric itself. For the P@k metric, a value equal or greater than 50% was considered ideal, given the dataset is unbalanced with very few positive labels (675 out of 316,800), meaning that only 0.22% of the dataset contains positive labels (equal to 1).

Initially, a 5-fold cross-validation was conducted. Since the dataset contains 675 positively labeled items, it is expected that each fold will have $675/5 = 135$ positive labels. Therefore, values of k between 100 and 150 were chosen accordingly. For k=100, we obtain a P@k metric equal to 0.63 with a standard deviation of 0.06. For k=150, the metric decreases as expected, but it reach a value of 0.55, with standard deviation 0.03, while staying within the established goal.

With the model evaluation completed, we proceeded to the validation phase of the model. As mentioned earlier, a Bayesian validation was conducted to choose the model that best suited the problem at hand. At this point, two experiments were performed. The first experiment included the Decision Tree model with parameters max depth=2 and random state=0

as a baseline. For comparison, the Random Forest models with parameters n estimators=10 and random state=0 was used, along with the CatBoost model properly calibrated using a threshold of 0.6 and the parameters: iterations=500, depth=6, and learning rate=0.1.

A Region of Practical Equivalence (ROPE) of 0.01 was used in each of the experiments, where if the posterior distributions of the models lie within the interval $[-0.01, +0.01]$, it can be stated that the models would be equivalent to the baseline model. If the posterior falls below this interval, the model will be considered less suitable than the baseline for modeling the situation. If it falls above, the model will be more suitable for modeling the problem.

The graphs of the posterior distributions are presented in Figure 7. As observed, both the CatBoost and Random Forest models are more suitable than the Decision Tree for modeling the problem, as they are entirely situated within the interval $[0.035-0.065]$. It is noted that both models are appropriate for use in the problem at hand.

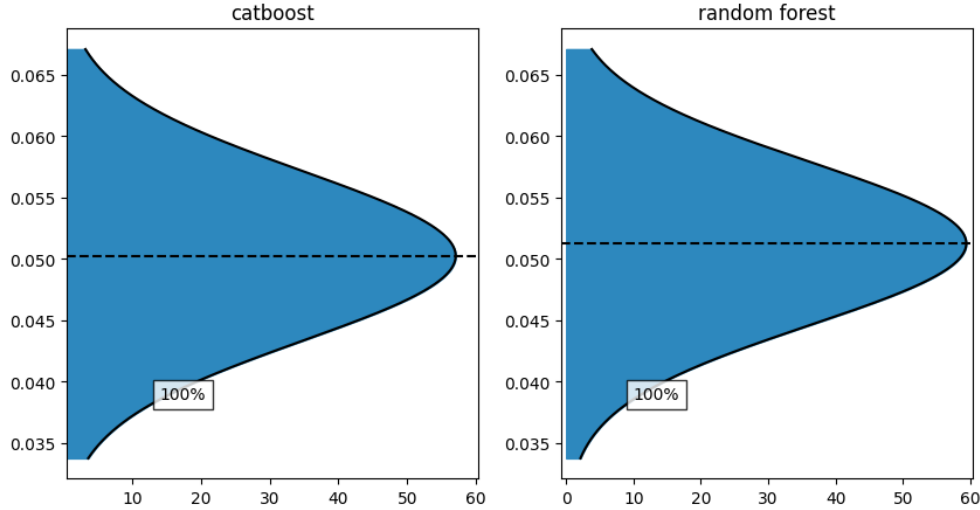


Figure 7: Model search result. Experiment 1. Source: Authors.

The second experiment involved replacing the baseline with the CatBoost model. As can be seen in figure 8, the Random Forest model (green distribution) is almost entirely within the interval $[-0.01 - +0.01]$, falling within the ROPE. Therefore, the conclusion obtained in the previous experiment is reaffirmed, that the CatBoost and Random Forest models are equivalent. Additionally, it is quite evident that the Decision Tree is less suitable for

modeling the problem than these two, as its posterior distribution lies entirely below the posterior distributions of the other models.

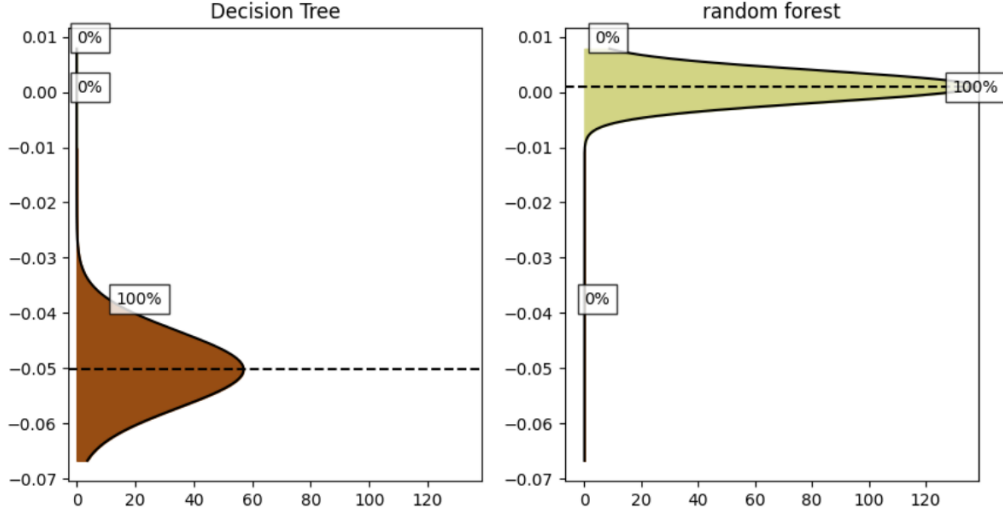


Figure 8: Model search result. Experiment 2. Source: Authors.

Since the Random Forest and CatBoost Regression models were considered equivalent, the decision was made to use the CatBoost model, as it handles categorical variables and provides an output value between 0 and 1, as mentioned early. This output value can later be transformed into a classification model by defining a threshold, ensuring greater flexibility for the model.

Once the original dataset contained 54 features, it was necessary to exclude some that were unnecessary in order to simplify the questions that the user would have to answer in the frontend. Therefore, as mentioned, a Bayesian feature validation was conducted using the same library.

In this Bayesian feature validation, a feature importance analysis was initially conducted to identify which features had the greatest impact on the model. The results of the feature importance analysis can be observed in figure 9.

After the feature importance graph was created, a correlation matrix between the features was developed. The intersections between the features that are white indicate a perfect correlation between such features (Pearson correlation coefficient equal to 1). Therefore, one of the two correlated features can be eliminated. The correlation matrix is shown in the figure 10.

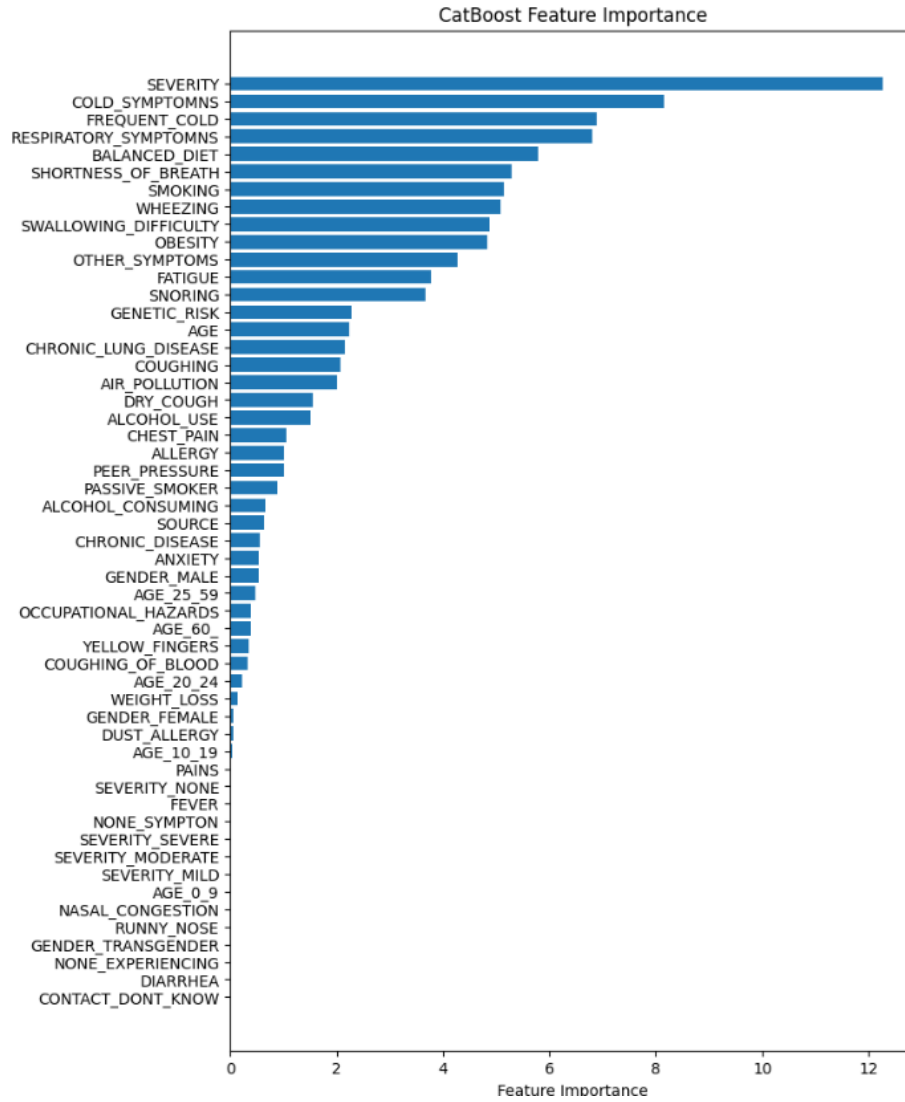


Figure 9: Feature importance analysis. Source: Authors.

As can be observed from the correlation matrix, there are several white areas where the features have a perfect correlation. In this case, if two features have perfect correlation (i.e., the correlation coefficient is equal to 1), one of them can be dropped. Thus, many features can be dropped. In addition to the model containing all 54 features, the following models were suggested:

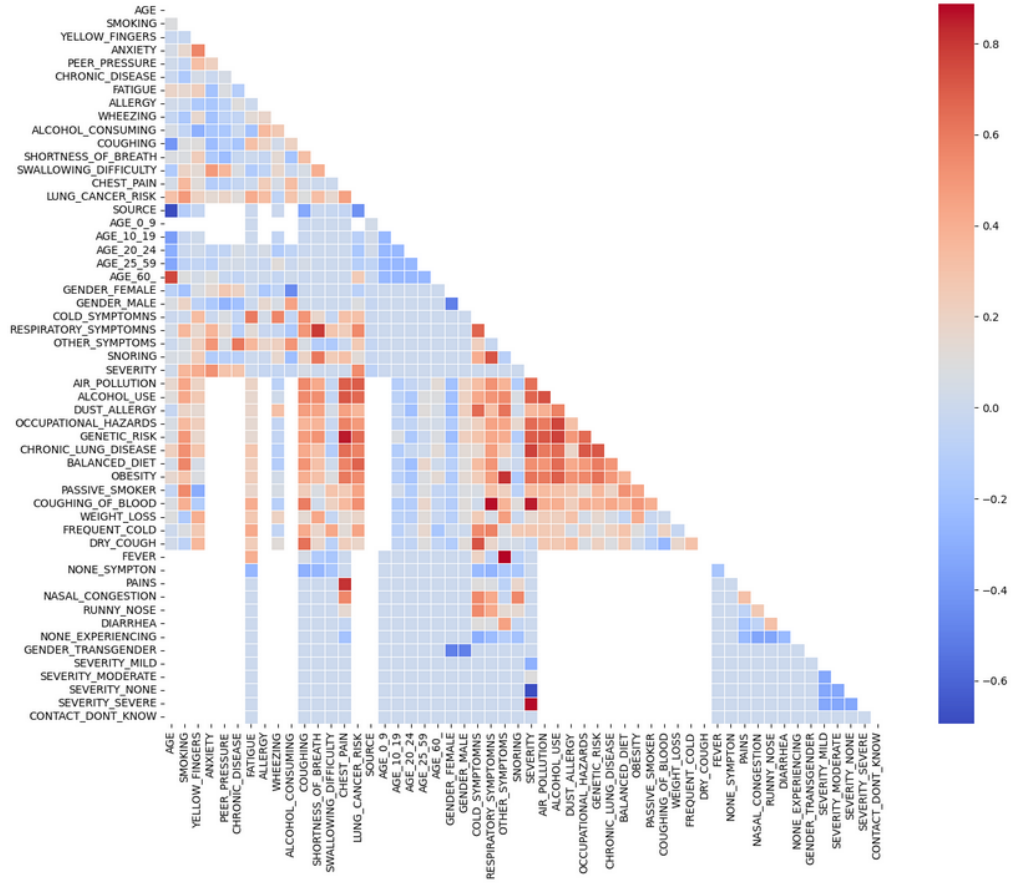


Figure 10: Correlation matrix. Source: Authors.

- A model with 14 features dropped.
- Another model with all the 14 features above dropped plus 4 features dropped too, totalizing 18 features dropped.
- Finally, a fourth model was developed for control purposes, where the 9 features considered most important in the feature importance (figure 9) were dropped.

The list of feature is available in appendix, table 5. The Bayesian validation of the models was then conducted by setting the model containing all the features (Model 1) as the baseline. The posterior distributions for Models 2, 3, and 4 are shown below:

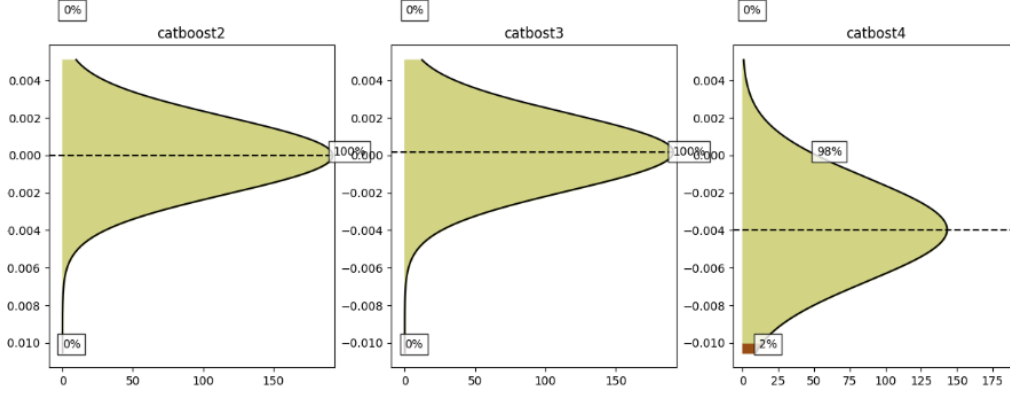


Figure 11: Feature selection with Bayesian validation results. Source: Authors.

From the graphs shown, it can be observed that Models 2 and 3 are equivalent to Model 1, which contains all the features. Therefore, Model 3 is chosen as it contains fewer features than the others. Model 4 is still viable but shows inferior performance compared to the other three models, which is entirely plausible since the most important features were dropped, serving as a control parameter for the experimentation.

Once the model is chosen, it is necessary to calibrate its parameters, especially the threshold to be used for converting the regression model into a classification model. For this purpose, various thresholds were tested, and graphs were plotted showing the percentages of individuals labeled as 0 or 1.

According to Dai et al. (2022), approximately 1.18 billion people worldwide are smokers, which is one of the risk factors for developing lung cancer, among others. Considering a global population of over 8 billion, the percentage of smokers in the population is about 13%. It is clear that not all smokers will develop lung cancer over their lifetime.

Additionally, the percentage of the population that develops cancer at some point in their lives is about 6%, according Lung Cancer Research Foundation- LCRF (2023). Therefore, the percentage of people with a high risk of developing lung cancer at any given time should be below 6%. Thus, a percentage of items labeled as high risk for developing lung cancer between 2-4% was considered ideal for this study.

Based on this, the threshold was calibrated so that the final model labels about 2 to 4% of individuals as 1. The threshold set at 0.7 gives a proportion of 2.3 % of individuals with a high risk of developing lung cancer, as shown

in figure 12, which is considered ideal.

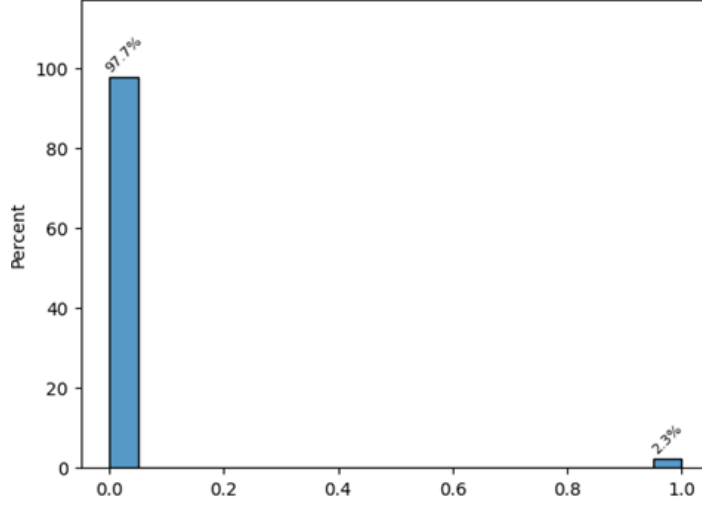


Figure 12: The final result generated by the model is as follows. Source: Authors.

After selecting and calibrating the model, a new need was identified to conduct feature importance in order to select the 15 most important variables. The questions in the frontend were formulated and ordered based on the knowledge of these variables. The new feature importance is shown in figure 13.

7. Discussion

At this point, we can enumerate the strengths and weaknesses in each phase of project implementation. Regarding data handling and data integration, the main vulnerabilities found were the lack of exploration of each variable's nature, which could be done in the future in collaboration with healthcare professionals, and the uncertainty present in survey data. In the future, we could incorporate laboratory results as features to include more continuous variables.

The main strengths of this phase were that it was possible to create a suitable input for the model to process the data satisfactorily, the approach used the most rigorous methodologies, and finally, it considered the possibility of using different datasets as inputs.

In the model search phase, the main strengths were that Bayesian validation and cross-validation provided the model with greater robustness and

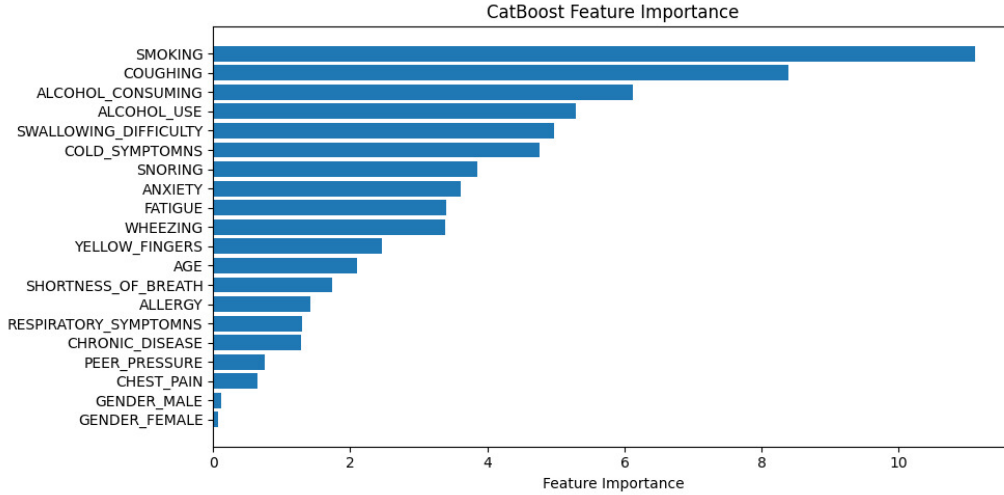


Figure 13: Feature importance for the frontend design questions. Source: Authors.

reliability. The Catboost Regression model has more flexibility than the Catboost Classifier model, meaning it is more suitable for handling missing data. Finally, it was found that feature selection discards unnecessary features during model development, simplifying the questions that the user must ask the patient in the frontend.

On the other hand, Bayesian validation consumes a lot of machine execution time and memory, as it requires repeating the experiment 10 times with 10 folds. Another weakness, which was overcome, was that the dataset, due to having many missing values, made the use of traditional metrics impractical, leading to the use of the $P@k$ metric, which is somewhat more difficult to implement and not as intuitive as others.

Looking at the application as a whole, we can affirm that the primary strengths are derived from a resilient development workflow, incorporating continuous integration and continuous deployment (CI/CD), along with rigorous validation through Bayesian methods and evaluation using appropriate metrics for imbalanced datasets. However, our weaknesses include the binary nature of the features, which could benefit from a more nuanced representation, such as ordinal levels. Additionally, enhancing the handling of missing data could improve our results. These issues could be addressed in a subsequent expansion of this work.

8. Conclusion

In light of the above, we can observe that available Machine Learning tools, coupled with software development tools, can yield highly useful data science projects. In this study, it was observed that such tools, when combined appropriately, can generate software capable of selecting patients at high risk of developing lung cancer, using a scientific methodology rather than relying solely on healthcare professionals' intuition.

Therefore, it is expected that when a patient at high risk of developing lung cancer is detected, they will be referred to an oncologist for further tests to confirm or rule out lung cancer. In this regard, the software's learning potential using feedback becomes evident, as healthcare professionals can mark a button indicating whether the patient indeed developed cancer.

9. Appendix

9.1. Dataset 1

#	Feature	Description
1	GENDER	M(male), F(female)
2	AGE	Age of the patient
3	SMOKING	YES=2, NO=1
4	YELLOW_FINGERS	YES=2, NO=1
5	ANXIETY	YES=2, NO=1
6	PEER_PRESSURE	YES=2, NO=1
7	CHRONIC_DISEASE	YES=2, NO=1
8	FATIGUE	YES=2, NO=1
9	ALLERGY	YES=2, NO=1
10	WHEEZING	YES=2, NO=1
11	ALCOHOL	YES=2, NO=1
12	COUGHING	YES=2, NO=1
13	SHORTNESS_OF_BREATH	YES=2, NO=1
14	SWALLOWING_DIFFICULTY	YES=2, NO=1
15	CHEST_PAIN	YES=2, NO=1
16	LUNG_CANCER	YES, NO

Table 2: Features and their descriptions

9.2. Dataset 2

Nr	Column name	Description
1	AGE	The age of the patient.
2	GENDER	The gender of the patient.
3	AIR_POLLUTION	The level of air pollution exposure of the patient.
4	ALCOHOL_USE	The level of alcohol use of the patient.
5	DUST_ALLERGY	The level of dust allergy of the patient.
6	OCCUPATIONAL_HAZARDS	The level of occupational hazards of the patient.
7	GENETIC_RISK	The level of genetic risk of the patient.
8	CHRONIC_LUNG_DISEASE	The level of chronic lung disease of the patient.
9	BALANCED_DIET	The level of balanced diet of the patient.
10	OBESITY	The level of obesity of the patient.
11	SMOKING	The level of smoking of the patient.
12	PASSIVE_SMOKER	The level of passive smoker of the patient.
13	CHEST_PAIN	The level of chest pain of the patient.
14	COUGHING_OF_BLOOD	The level of coughing of blood of the patient.
15	FATIGUE	The level of fatigue of the patient.
16	WEIGHT_LOSS	The level of weight loss of the patient.
17	SHORTNESS_OF_BREATH	The level of shortness of breath of the patient.
18	WHEEZING	The level of wheezing of the patient.
19	SWALLOWING_DIFFICULTY	The level of swallowing difficulty of the patient.
20	CLUBBING_OF_FINGER_NAILS	The level of clubbing of finger nails of the patient.

Table 3: Column names and their descriptions

9.3. Dataset 3

#	Column Name
1	FEVER
2	TIREDNESS
3	DRY_COUGH
4	DIFFICULTY_IN_BREATHING
5	SORE_THROAT
6	NONE_SYMPTON
7	PAINS
8	NASAL_CONGESTION
9	RUNNY-NOSE
10	DIARRHEA
11	NONE_EXPERIENCING
12	AGE_0_9
13	AGE_10_19
14	AGE_20_24
15	AGE_25_59
16	AGE_60+
17	GENDER_FEMALE
18	GENDER_MALE
19	GENDER_TRANSGENDER
20	SEVERITY_MILD
21	SEVERITY_MODERATE
22	SEVERITY_NONE
23	SEVERITY_SEVERE
24	CONTACT_DONT-KNOW
25	CONTACT_NO
26	CONTACT_YES
27	COUNTRY

Table 4: Column names

col	feature	ds1	ds2	ds3	m1	m2	m3	m4
1	AGE	p	p	p	p	p	p	p
2	SMOKING	p	p	a	p	p	p	p
3	YELLOW_FINGERS	p	r	p	p	p	p	p
4	ANXIETY	p			p			
5	PEER_PRESSURE	p			p	p	p	p
6	CHRONIC_DISEASE	a			p	p	p	p
7	FATIGUE	p	p	r	p	p	p	p
8	ALLERGY	p			p	p	p	p
9	WHEEZING	p	p		p	p	p	p
10	ALCOHOL_CONSUMING	a			p	p	p	p
11	COUGHING	p	a	r	p	p	p	p
12	SHORTNESS_OF_BREATH	a	p	r	p	p	p	p
13	SWALLOWING_DIFFICULTY	a	p	r	p	p	p	p
14	CHEST_PAIN	a	p	a	p	p	p	p
15	LUNG_CANCER_RISK	r	r		p	p	p	p
16	GENDER	p	p		p	p	p	p
17	COLD_SYMPTOMNS	a	a	a	p	p	p	
18	RESPIRATORY_SYMPTOMNS	a	a	a	p	p	p	
19	OTHER_SYMPTOMS	a	a	a	p	p	p	
20	SNORING	a	p	a	p	p	p	p
21	SEVERITY	a	a	a	p	p	p	p
22	AIR_POLLUTION			p				
23	ALCOHOL_USE			p	p	p	p	p
24	DUST_ALLERGY			p				p
25	OCCUPATIONAL_HAZARDS			p	p	p	p	p
26	GENETIC_RISK			p				
27	CHRONIC_LUNG_DISEASE			p	p	p	p	p
28	BALANCED_DIET			p				
29	OBESITY			p	p	p		p
30	PASSIVE_SMOKER			p				
31	COUGHING_OF_BLOOD			p	p	p	p	p
32	WEIGHT_LOSS			p				
33	FREQUENT_COLD			p	p	p	p	p
34	DRY_COUGH		p	p				
35	FEVER			p				
36	NONE_SYMPTON				p	p	p	
37	PAINS			p			p	p
38	NASAL_CONGESTION			p		p		p
39	RUNNY_NOSE		p	p		p		p
40	DIARRHEA			p			p	p
41	NONE_EXPERIENCING			p			p	p
42	SEVERITY	a	a	a	p	p	p	p
43	CONTACT				p			p
44	COUNTRY		p	p				

Table 5: Legend for Integrated Dataset Features: **p** for present, **a** for artificial, **r** for renamed, **ds** for dataset, **m** for model

References

- 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.
- Bitencourt, W.A., Silva, D.M., Xavier, G.d.C., 2022. Pode a inteligência artificial apoiar ações contra evasão escolar universitária? Ensaio: Avaliação e Políticas Públicas em Educação 30, 669–694. URL: <http://dx.doi.org/10.1590/S0104-403620220003002854>, doi:10.1590/s0104-403620220003002854.
- Chandra, U., Reps, J., Yang, R., Vachani, A., Maldonado, F., I., K., 2023. Machine learning and real-world data to predict lung cancer risk in routine care. *Cancer Epidemiology Biomarkers Prev* 32, 337–343.
- Corani, G., Benavoli, A., 2015. A bayesian approach for comparing cross-validated algorithms on multiple data sets. *Machine Learning* 100(2), 285–304. doi: 10.1080/s10994-015-5486-z.
- Dai, X., E, G., D., L.A., 2022. Evolution of the global smoking epidemic over the past half century: strengthening the evidence base for policy action. *Tob Control* 31(2), 129–137 doi: 10.1136/tobaccocontrol-2021-056535.
- Dong, G., Liu, H., 2018. Feature Engineering for Machine Learning and Data Analytics. Chapman & Hall / CRC data mining & knowledge discovery series, Taylor & Francis Group. URL: <https://books.google.com.br/books?id=eYqvswEACAAJ>.
- LCRF, 2023. Facts about lung cancer. <https://www.lungcancerresearchfoundation.org/lung-cancer-facts>.
- PAHO, 2024. Global cancer burden growing, amidst mounting need for services. <https://www.paho.org/en/news/1-2-2024-global-cancer-burden-growing-amidst-mounting-need-services>.
- Quanyang, W., Yao, H., Sicong, W., Linlin, Q., Zewei, Z., Donghui, H., Hongjia, L., Shijun, Z., 2024. Artificial intelligence in lung cancer screening: Detection, classification, prediction, and prognosis. *Cancer Med.* 13(7), e7140. doi: 10.1002/cam4.7140.

- Richter, A.N., Khoshgoftaar, T.M., 2018. A review of statistical and machine learning methods for modeling cancer risk using structured clinical data. *Artificial Intelligence in Medicine*. 90 Issue C, 1–14.doi:10.1016/j.artmed.2018.06.002.
- Silva, V.S., Campos, M.L.M., Silva, J.C.P., Cavalcanti, M.C., 2011. An approach for the alignment of biomedical ontologies based on foundational ontologies. *Journal of Information and Data Management* 2, 557–557.
- Vapnik, V.N., 1999. *The nature of statistical learning theory*. 2nd ed., Springer-Verlag, New York.
- Verri, F.A.N., 2024. *Data Science Continuum: From Foundations to Practices*. Pre-print ed.
- WHO, 2024. Cancer. <https://www.who.int/news-room/fact-sheets/detail/cancer>.