Application of Machine Learning in Tuberculosis Data from the Brazilian Public Health System

Miguel Pereira Ramos[1,2], Rodrigo Bonacin[1], Ana Carolina Borges Monteiro Padilha[1], Reinaldo Padilha França[1]

mpramos@cti.gov.br, rodrigo.bonacin@cti.gov.br, ana.monteiro@cti.gov.br, reinaldo.franca@cti.gov.br

**[1]Divisão de Metodologias da Computação – DIMEC
CTI/MCTI Renato Archer – Campinas/SP**

**[2]Instituto de Computação
Universidade Estadual de Campinas – Campinas/SP**

Abstract. This research explores the application of machine learning in the public health supply chain, focusing on tuberculosis in Brazil. Using Unified Health System (SUS) data, it proposes predictive models to identify outbreak patterns, predict the need for medicine and supply demand, optimize logistics, and reduce shortages. The analysis employs supervised algorithms like XGBoost, trained on epidemiological, demographic, and logistics data. Insights from these models can integrate into real-time decision support systems for supply chain managers. The paper also discusses infrastructure, data interoperability, and privacy challenges, alongside expected benefits in efficiency, patient safety, and system sustainability. This work demonstrates how digital transformation, facilitated by artificial intelligence, can enhance public health, offering a replicable model for developing countries.

1. Introduction

Tuberculosis (TB) is an infectious disease caused by Mycobacterium tuberculosis, a bacterium that more commonly affects the lungs. It is spread through the air with droplets of saliva when people with TB cough, sneeze, or spit. TB remains a major global health challenge, being one of the main causes of death from a single infectious agent and affecting all age groups and countries across the globe [1][2].

The treatment of TB consists of special antibiotics intake for 4-6 months. Those drugs must be taken daily, and their early stop can lead to drug resistance, configuring drug-resistant tuberculosis (DR-TB), requiring special treatment with different medicines [1]. This highlights the critical role of medicine availability in TB treatment, and the danger of missing drug intake. This points to the focus of this research, which aims to increase the trust in the public health Supply Chain Management (SCM), with special emphasis on medicines and rapid tests.

Despite being possible to approach these problems with other traditional methods, such as data analysis and other ways to solve SCM problems, ML provides an efficient way to process large volumes of data and to analyze the problem, while being an area with crescent development.

Therefore, a modern way to solve this problem, with more research being done every year, is the use of machine learning for predictive analysis. In this paper, AI was used to build a predictive approach for SCM with clinical and demographic data.

2. Overview of Tuberculosis

Although TB is curable and treatable, it remains a great challenge for public health in many countries, including Brazil, being one of the main causes of mortality and morbidity. After 2023, the World Health Organization (WHO) stated that TB came back to be the main cause of death from a single agent, surpassing COVID-19, and in the same year, it was estimated that 10.8 million people got sick from TB, and 1.25 million died because of it [3].

In the year of 2024, 654,824 rapid molecular tests were done to diagnose TB in Brazil, and 84,308 new cases were notified, while 15.3% of 2023 cases had LTFU as outcome, and 1,047 new cases of DR-TB were identified. To react to these numbers, R$100,000,000 was incremented in the budget destined to fight TB across the country. This resource distribution took into consideration the percentage of new cases notified in each federation unit in 2022 [4].

That way, the management of this kind of resource, the distribution, and the efficiency of both financial and logistical infrastructure can be promoted and improved through the support of predictive models. This way, it is possible to reduce medical supply waste, be prepared, in advance, for its necessity, and avoid misused financial resources.

3. ML Literature Overview

One of the most powerful "family" of models for classification with tabular data are the Decision Tree models [5][6]. They use a hierarchy tree structure, where every internal node represents a functionality, a branch represents a decision rule, and each leaf node represents the result of the processing. Naturally, decision trees are prone to overfitting, which led to the development of strategies to counter it, such as Random Forests and Boosting techniques.

Random Forest models utilize randomness to select characteristics with low correlation between each tree, so that the resulting forest has low correlation when training, reducing the overfit. On the other hand, models that use Boosting solve this problem by combining many weak individual trees, that is, models that show little improvement from randomness, to build a strong tree. Each weak tree is trained to correct the errors made by previous models. After enough iterations, the weak trees are promoted to strong trees [7][8].

A specific type of Boosting, which is the one that will be used in the study, is the Gradient Boosting, a Boosting algorithm that uses the Gradient Descent. Gradient Descent is an optimization algorithm used to train ML models, commonly employed to train Neural Network models. Briefly, it represents a way to measure the numerical error between the model's predicted result and reality, leading the model to minimize this error. With the training data, the model learns with each training iteration and descends in the gradient. Until the gradient function is almost or equal to zero, the model adjusts its parameters to obtain the lowest error possible.

It cannot be left out of the explanation of the cross-validation technique. It is a statistical method to evaluate and compare a given ML model that consists of partitioning a dataset into training and validation data. It differentiates itself from the common data splitting technique because every portion of data is used for testing and every portion is used for validation, while the basic split of training and testing data is fixed.

The common cross-validation creates crossover in different rounds of testing, so that any given entry has a chance to be validated against. The typical cross-validation method, used in this study, is the k-fold cross-validation.

In k-fold cross-validation, the data is separated into k equally sized folds. Then, k iterations are performed with a different fold held out for validation, while the other k - 1 are used for training. The result of each iteration is then analyzed using a chosen metric, in a way that more homogenous numbers are good indicators of model robustness and that it is not highly sensitive to training data [9].

Cross-validation is important to evaluate a model's performance against new or unseen data and to check for overfitting, that is, being too dependent on the data used to train it. It can also be used to model selection, being used to have more trustworthy metrics for comparison and for tuning models to a specific dataset.

4. Methodology ML

Figure 1 presents the methodological flowchart employed in this study. The diagram briefly shows the procedure for obtaining data from the SINAN, followed by feature selection, data preprocessing, and partitioning into training and testing datasets. Subsequently, performance metrics and graphical representations are utilized to evaluate the results and identify the optimal model. The chosen model is then submitted to cross-validation and SHAP analysis.

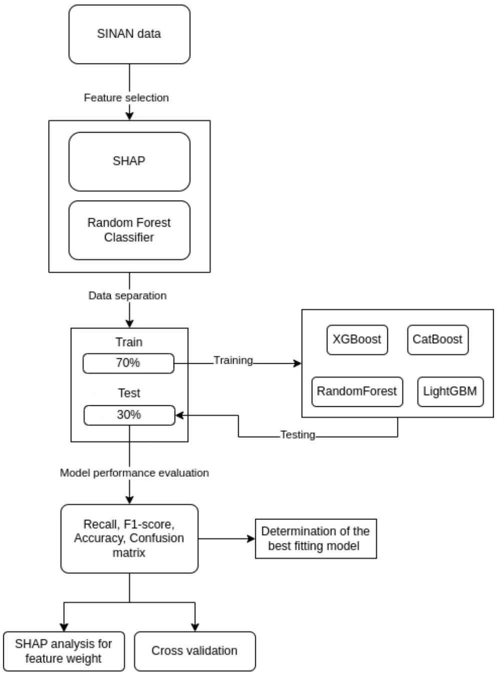


Figure 1. Methodology flowchart

4.1. Environment and Libraries

A Jupyter Notebook in Visual Studio Code was used as the environment to develop this study. The libraries imported to the script included Numpy for mathematical operations and array handling, Matplotlib and Seaborn for data visualization, Pandas for data manipulation and analysis, ImbalancedLearn for handling under-sampling and oversampling, and Sklearn for training-testing split, metrics, and confusion-matrix generation, cross-validation, and One vs. Rest classification.

4.2. Data preprocessing

Data was loaded from SINAN (Notifiable Diseases Information System), which is part of the DATASUS platform maintained by the Brazilian Ministry of Health, with the help of the library PySUS. This data contains TB case notifications from the years 2001 to

2024 that hold demographic and clinical information of patients and their outcomes. SINAN is an essential tool of the Brazilian government, responsible for collecting, processing, and distributing data on diseases across the country. Each notification contains demographic (age, race, sex, etc) and clinic (test results, like sputum smear microscopy) information from the patient.

Thus, each row contains information about a specific patient, and each column represents one of the many important pieces of information. This way, the final status column of the notification was, then, used as the y column, so that the model should predict it. The rows without this information were removed, and the dataset was further cleaned. Also, the values were converted to integers that represented a class for each feature, as well as the "final status" column values.

Before proceeding to manipulate features and find the most relevant ones, features without any value, like identification numbers, process dates, and others, were also removed from the dataset, and the columns were renamed to improve legibility. Then, the dataset was divided into testing and training data, with the first one being 30% of the complete dataset.

4.3. Feature selection

It was tested to submit the features to the Random Forest Classifier, for it to select the 20 most relevant ones, trying to lower noise from features that were not so important. Other tests were done with all the meaningful features. After this process, the dataset with the selected features was used to train and evaluate the models. These are XGBoost, CatBoost, LightGBM, and RandomForest.

4.4. Model training

Each model was submitted to One vs. One and One vs. Rest training, with both under-sampled and over-sampled data, because of the imbalance in the outcome classes. Afterward, the best results were submitted to training with 5-fold cross-validation, to check for overfitting or any bias.

4.5. Model Evaluation and explainability

Finally, to analyze the results, a SHAP explainer was created for the models with the best results, and a random sample of the training data with 100 rows was selected.

After passing the sample through the explainer, the SHAP data were plotted for each class. A confusion matrix for each model was also plotted, along with other metrics.

5. Results

The results of the experiments, done as described in the previous section, were evaluated in terms of the precision, recall, and F1-score of each class and the weighted average of each metric, as well as the accuracy of the general results. Other results used to analyze the quality of the model were the plotting of the confusion matrix of each model.

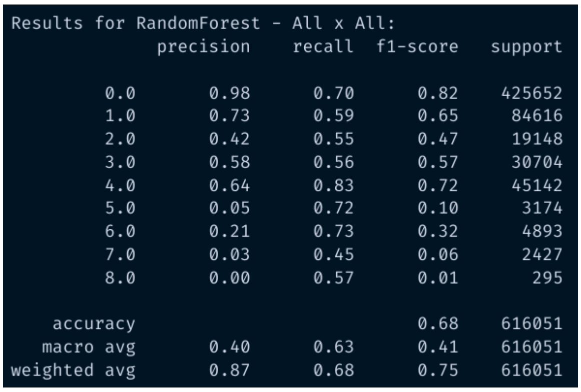
The classes predicted by the models are shown below in Table 1.

Table 1.Classes used in the classification

0	1	2	3	4	5	6	7	8
Cure	LTFU	Death by TB	Death by other causes	Transfer	Change in diagnosis	DR-TB	Scheme change	Failure

5.1. Random Forest

The Random Forest model, the only model tested that does not use boosting, using undersampled data, achieved 87% weighted precision, 68% weighted recall, and 75% weighted F1-score. Figure 2 shows the complete report of the results.

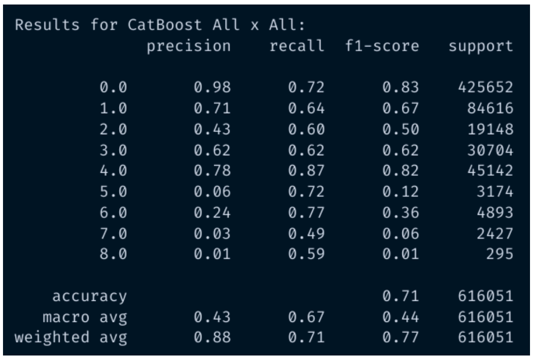


```
Results for RandomForest - All x All:
              precision    recall  f1-score   support

         0.0       0.98      0.70      0.82    425652
         1.0       0.73      0.59      0.65     84616
         2.0       0.42      0.55      0.47     19148
         3.0       0.58      0.56      0.57     30704
         4.0       0.64      0.83      0.72     45142
         5.0       0.05      0.72      0.10      3174
         6.0       0.21      0.73      0.32      4893
         7.0       0.03      0.45      0.06      2427
         8.0       0.00      0.57      0.01       295

    accuracy                           0.68    616051
   macro avg       0.40      0.63      0.41    616051
weighted avg       0.87      0.68      0.75    616051
```

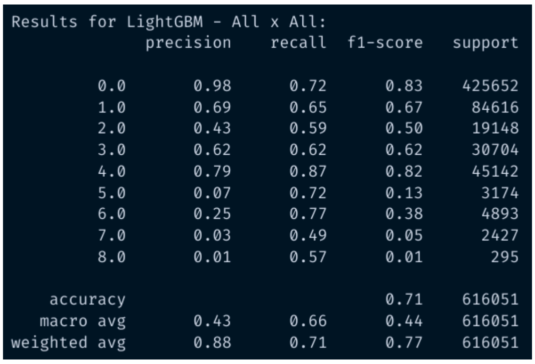
Figure 2: Random Forest One vs. One metrics

5.2 Catboost and Light GBM

Catboost and LightGBM had similar results (Figures 3 and 4), obtaining 88% of final weighted precision, 71% weighted recall, and 77% weighted F1-score. Catboost achieved 65.91% in f1-weighted mean in cross-validation.



```
Results for CatBoost All x All:
              precision    recall  f1-score   support

         0.0       0.98      0.72      0.83    425652
         1.0       0.71      0.64      0.67     84616
         2.0       0.43      0.60      0.50     19148
         3.0       0.62      0.62      0.62     30704
         4.0       0.78      0.87      0.82     45142
         5.0       0.06      0.72      0.12      3174
         6.0       0.24      0.77      0.36      4893
         7.0       0.03      0.49      0.06      2427
         8.0       0.01      0.59      0.01       295

    accuracy                           0.71    616051
   macro avg       0.43      0.67      0.44    616051
weighted avg       0.88      0.71      0.77    616051
```



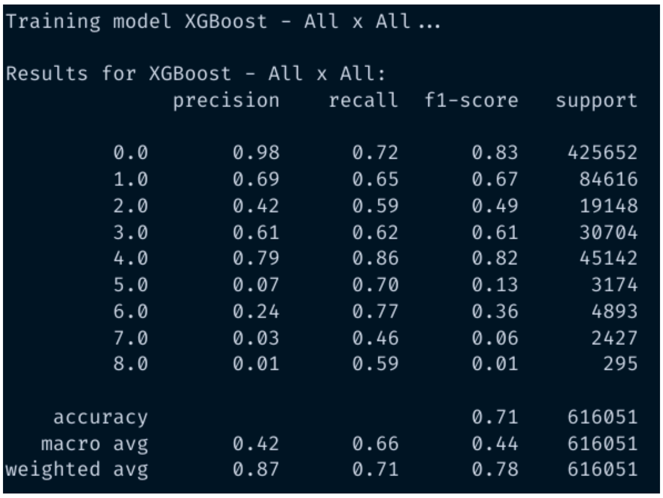
```
Results for LightGBM - All x All:
              precision    recall  f1-score   support

         0.0       0.98      0.72      0.83    425652
         1.0       0.69      0.65      0.67     84616
         2.0       0.43      0.59      0.50     19148
         3.0       0.62      0.62      0.62     30704
         4.0       0.79      0.87      0.82     45142
         5.0       0.07      0.72      0.13      3174
         6.0       0.25      0.77      0.38      4893
         7.0       0.03      0.49      0.05      2427
         8.0       0.01      0.57      0.01       295

    accuracy                           0.71    616051
   macro avg       0.43      0.66      0.44    616051
weighted avg       0.88      0.71      0.77    616051
```

Figure 3: Catboost One vs. One metrics

Figure 4: LightGBM One vs. One metrics

5.3 XGBoost

Finally, XGBoost got an overall accuracy of 87%, 71% weighted recall, and 78% weighted F1-score. The One vs. One model achieved an 81% F1-weighted mean in the cross-validated results using over-sampled data. Figure 5 shows the complete report, Figure 6 shows the confusion matrix for XGBoost, and Figure 7 shows the cross-validated report.



```
Training model XGBoost - All x All...

Results for XGBoost - All x All:
              precision    recall  f1-score   support

         0.0       0.98      0.72      0.83    425652
         1.0       0.69      0.65      0.67     84616
         2.0       0.42      0.59      0.49     19148
         3.0       0.61      0.62      0.61     30704
         4.0       0.79      0.86      0.82     45142
         5.0       0.07      0.70      0.13      3174
         6.0       0.24      0.77      0.36      4893
         7.0       0.03      0.46      0.06      2427
         8.0       0.01      0.59      0.01       295

    accuracy                           0.71    616051
   macro avg       0.42      0.66      0.44    616051
weighted avg       0.87      0.71      0.78    616051
```

Figure 5: XGBoost One vs. One metrics

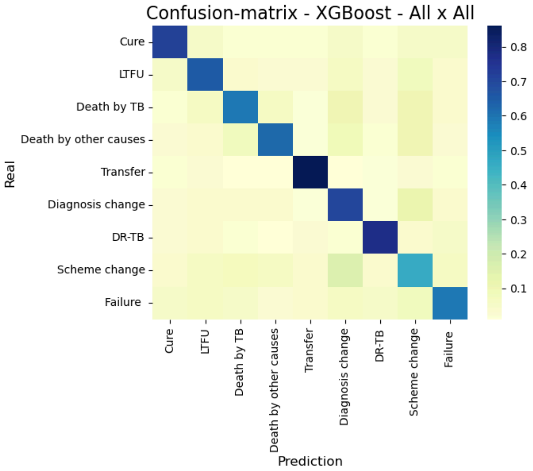
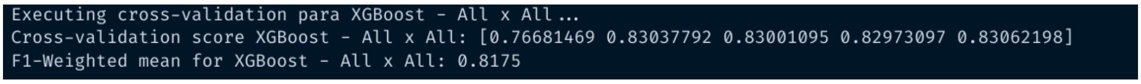


Figure 6: Confusion Matrix for One vs. One XGBoost



```
Executing cross-validation para XGBoost - All x All...
Cross-validation score XGBoost - All x All: [0.76681469 0.83037792 0.83001095 0.82973097 0.83062198]
F1-Weighted mean for XGBoost - All x All: 0.8175
```

Figure 7: Cross-validation results for One vs. One XGBoost

5.4 SHAP Explanation

Before looking at the results of SHAP analysis and graphical representations, it is important to clarify how SHAP works and how the beeswarm represents the influence of each feature on a model's prediction.

SHAP stands for Shapley Additive exPlanations. SHAP values are obtained through a process of importance evaluation that represents the contribution of each feature to the final result. It uses a game-theoretical model to find these contributions, where each feature represents an agent that interferes with the model's processing. Also, SHAP unifies 7 different methods in a framework, providing robustness to the results of the explanation.

More than that, the SHAP framework provides a graphical visualization of the results, be it a single prediction, with detailed influence of each feature on a specific prediction

or general information about feature weight for each class. Generally, SHAP relates feature values to the impact on model output.

That said, the three most relevant classes, which do not represent administrative results, were analyzed using beeswarm graphics. It represents the SHAP values on the horizontal axis and the most relevant features on the vertical axis, while representing the feature value through a color spectrum. This means that for a determined feature and determined class, a blue dot on the right extreme shows that lower values of that feature impacts strongly in the positive classification of that class, for example.

Figures 8, 9, and 10, shown below, are the beeswarm graphics for the three most important classes: LTFU, Cure, and Death by TB.

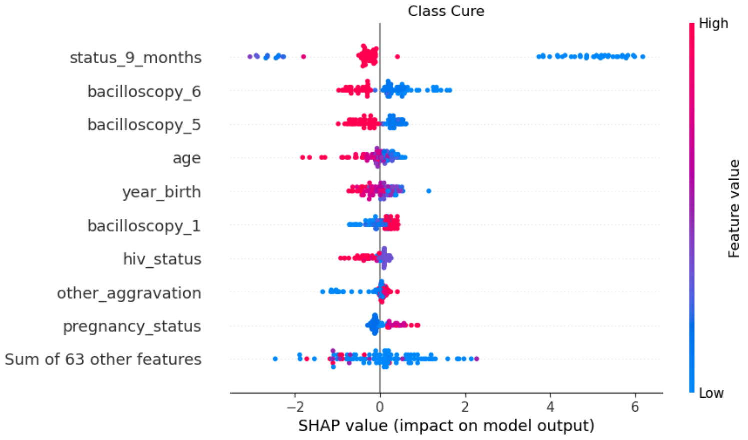


Figure 8: *beeswarm* graphics for Cure

Figure 8 shows the features that mostly contributed to the final result of the cure. The stronger features were *status_9_months* that indicates the situation of the patient after 9 months of treatment, *bacilloscopy_6, bacilloscopy_5, and bacilloscopy_1*, which indicates the sputum smear bacilloscopy result at the end of the sixth, fifth, and first month of treatment, respectively, *age,* the age of the patient when the notification was made, *year_birth, hiv_status, other_aggravation,* that states if the patient has other clinical aggravation, and *pregnancy_status*.

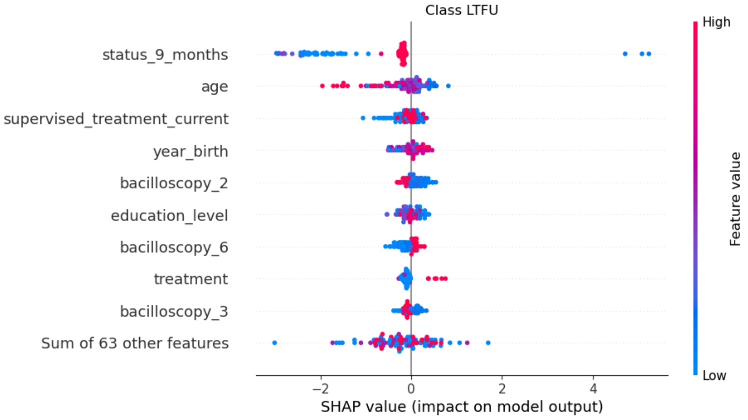


Figure 9: *beeswarm* graphics for Loss To Follow Up

Figure 9 shows the features that were more determinant for the LTFU class. The main features were status_9_month, age, supervised_treatment_current, which states if the patient did the supervised treatment until its end, year_birth, bacilloscopy_2, education_level, bacilloscopy_6, treatment, which states the type of treatment (if it is the first time or recurrent, for example), and bacilloscopy_3.

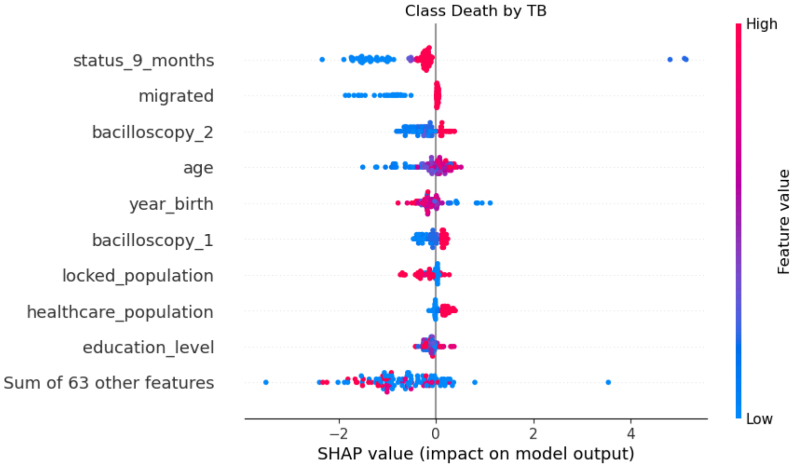


Figure 10: *beeswarm* graphics for Death by Tuberculosis

Finally, Figure 10 encloses the information about the features with greater influence on the result for Death by TB. The most important features were *status_9_months, migrated,* that states if the patient is a person that migrated to Brazil, *bacilloscopy_2, age, year_birth, bacilloscopy_1, locked_population,* that states if the patient is part of the special population that were in deprivation of liberty, *healthcare_population,* that states if the patient is in the special population that is a healthcare professional, and *education_level,* that states the level of education of the patient (complete second grade, for example).

6. Discussion

In all three results, the main contribution to the result was the feature that stated the status of the patient in the first nine months of treatment. This could mean that the TB treatment is stable once it is maintained for the established period. The age feature was also mandatory for all three results. It can be perceived that lower values in this field push negatively to Death by TB, while higher values are somewhat neutral. For Cure and LTFU, on the other hand, greater ages push negatively for these results. This indicates that younger people are less commonly associated with Death results, while older people are less commonly associated with Cure and LTFU results.

Another very important and recurrent field was that of bacilloscopy, that is, the sputum smear bacilloscopy results, of various months. This enlightens the importance of this test, which indicates the presence of the pathogenic agent in the patient’s body.

For the class Cure, it was possible to identify some features with high impact on the result related to physical conditions of the patient: pregnancy_status, hiv_status, and other_aggravation. This brings attention to the risks of comorbidities, pregnancy, and concurrent hiv infection for the positive response of the patient to the treatment.

For the classes of Death by TB and LTFU, on the other hand, the presence of social and demographic features was among the greatest. The feature education_level in the LTFU class enlightens the importance of the conscientization of the population, and more specifically, the patient, about the necessity of the continuation of the treatment. While the features locked_population, migrated, and healthcare_population, in the Death by TB class, indicate a great risk for marginalized populations.

Therefore, the first point to consider to associate the model’s results with comprehensible and useful information to SCM is the meaning of each resulting class to logistic implications. Each result can be associated with personnel and material needs, thus indicating a demand in the supply chain.

The result for Cure, although meaning a controlled situation, still shows a continuous need for medicine that can be more regular and predictable. Any death result and failure on the treatment, on the other hand, indicates a greater need for attention, with more tests, medicines, and professional care.

Moreover, the results for LTFU are one of the most important to be considered, once that patients with this risk need considerably more attention from healthcare and social assistance professionals, requiring domiciliary visits and even social care. Another very important result is DR-TB, once that it represents the necessity for non-conventional medicine and specific care.

These insights can represent a path to migrate from a reactive supply chain, which requires supply in the lack of it, to a predictive supply chain, which requires supply before the lack of it. This improves not only financial efficiency but also quality of life and treatment for both patients and professionals. Also, it reduces the need to stock supplies for safety, preventing losses and discarding, and waste production.

These findings suggest that predictive models, when properly trained and interpreted, can act as intelligence layers within the public health supply chain. In the context of tuberculosis in Brazil, such integration may increase treatment success, prevent logistic disruptions, and ultimately improve patient outcomes.

7. Conclusions

Tuberculosis represents one of the main challenges in the fight against endemic diseases, and this is reflected in the public health system in many ways. This study concluded that a suitable solution to the problem, in the aspect of supply chain, is to apply machine learning concepts to create predictive models to rewire the system from a reactive supply management to a predictive one.

This could lead to improvements in patient treatment, reduce supply waste, manage resources more efficiently and faster, and to an overall better quality of the health system.

This study proved the efficiency of ML techniques for classification in health problems and the usage of boosting and decision tree models in the solution for tabular data classification problems. Also, it made clear the advantages to the use of using explainers, as SHAP, for the understanding of the prediction results.

Furthermore, the importance of AI in the contribution of social problems, quality of life, and, specifically to this study, in health system challenges is reassuring. That means that this type of research represents great academic and social importance, once that it leads to the improvement of society through ethical and responsible use of AI.

8. Acknowledgements

This study was financed by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). We thank the Ministry of Health of Brazil for the availability of DATASUS.

9. References

[1] World Health Organization (2025) “Global tuberculosis report 2024”.

[2] Chakaya, J., Khan, M., Ntoumi, F., Aklillu, E., Fatima, R., Mwaba, P., Kapata, N., Mfinanga, S., Hasnain, S.E., Katoto, P.D.M.C., Bulabula, A.N.H., Sam-Agudu, N.A., Nachega, J.B., Tiberi, S., McHugh, T.D., Abubakar, I. and Zumla, A. (2021) “Global Tuberculosis Report 2020 - Reflections on the Global TB burden, treatment

and prevention efforts", International Journal of Infectious Diseases, 113 (Suppl 1), S7-S12.

[3] World Health Organization (2024) "Global Tuberculosis Report 2023".

[4] Ministry of Health of Brazil. (2025) "Epidemiological Report – Tuberculosis 2025".

[5] Uddin, S. and Lu, H. (2024) "Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data", PLoS ONE, 19(4).

[6] Patel, H. H. and Prajapati, P. (2018) "Study and Analysis of Decision Tree Based Classification Algorithms", International Journal of Computer Sciences and Engineering, 6(10), 74–78.

[7] França, R. P., Bonacin, R. and Monteiro, A. C. B. (2025) "A machine intelligence model based on random forest for data related renewable energy from wind farms in Brazil", Computer Vision and Machine Intelligence for Renewable Energy Systems. Elsevier, p. 127-139.

[8] Friedman, J. H. (2001) "Greedy function approximation: A gradient boosting machine.", The Annals of Statistics, Ann. Statist. 29(5), 1189-1232.

[9] Bishop, C. M. (2006) "Pattern Recognition and Machine Learning", New York: Springer.