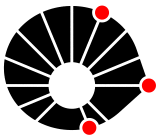

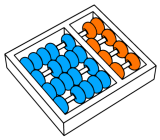

Cancer Survival Prediction through Multifactorial Analysis with Machine Learning

Authors:

Isabella Ribeiro Rigue *Luigi Mello Rigato*

Advisors:

Gabriel Bianchin de Oliveira *Zanoni Dias*

Technical Report - IC-PFG-26-23

Final Graduation Project

July 2026

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

Cancer Survival Prediction through Multifactorial Analysis with Machine Learning

Isabella Ribeiro Rigue Luigi Mello Rigato Zanoni Dias Gabriel Bianchin de Oliveira

July 2026

Abstract

Cancer remains a pressing public health challenge in Brazil, demanding data-driven strategies to improve patient outcomes. The objective of this work is to develop and evaluate machine learning and deep learning models to predict the overall survival of patients with cancer at the beginning of the initial treatment, using demographic, diagnostic, and treatment-related information available at that time. This study uses a large-scale dataset from the Brazilian National Cancer Institute (INCA). By processing over 5.4 million raw longitudinal records through a PySpark pipeline to resolve severe data quality and temporal inconsistencies, we extracted a refined cohort of 73,459 high-quality samples. Our predictive modeling explored binary classification (random forest, XGBoost, multi-layer perceptron, AutoGluon, and ensembles), achieving high discriminative power with ROC-AUC scores around 0.88, and advanced survival analysis (DeepSurv and XGBoost Survival) for time-to-event dynamics, where the DeepSurv neural network reached a C-index of approximately 0.85. Interpretability analyses revealed that the primary tumor site, metastasis staging, patient age, and treatment history are the most critical predictors of mortality risk. Ultimately, these results validate the feasibility of applying artificial intelligence to Brazilian hospital oncology records, demonstrating the potential of these models for future clinical decision-support applications within the Unified Health System (SUS).

1 Introduction

Cancer represents one of the most pressing public health challenges in Brazil, with a growing burden of incidence, mortality, and healthcare costs throughout the country [19]. The collective fight against this disease requires research across various scientific fields and must be supported by effective organization and data-driven insights [21, 34]. Therefore, since Brazil is a vast country with a large capacity for data generation, collaboration and research development should be encouraged. Population-based surveillance and timely access to diagnosis and treatment are essential to reduce avoidable deaths and inform resource allocation within the Brazilian Unified Health System (SUS). In this context, hospital cancer registries play a central role in epidemiological monitoring by systematically recording clinical, demographic, and treatment-related information for cancer patients over time [30, 20].

The Instituto Nacional de Câncer (INCA–Brazilian National Cancer Institute) coordinates cancer surveillance efforts in the country and maintains the *Registro Hospitalar de Câncer*[1], a structured source of longitudinal oncology records. Beyond descriptive epidemiology, historical registry data offer substantial potential for predictive analytics and evidence-based decision support, allowing the identification of survival patterns, treatment delays, and risk factors associated with adverse outcomes [27, 32, 22].

This study used a third-party Kaggle distribution derived from Brazilian Hospital Cancer Registry data coordinated by INCA [2]. The downloaded file contained approximately 5.4 million tumor-registry records dated from 1985 to 2023 and 47 original variables covering demographic, clinical, temporal, behavioral, and treatment-related information. To support reproducibility, we recorded the dataset version, download date, file checksum, and correspondence with the official registry data dictionary.

Despite its scale and richness, the dataset presents significant data-quality challenges. We identified a high rate of missing values in key staging variables, temporal inconsistencies among diagnosis, consultation, treatment, and death dates, categorically encoded fields requiring domain mapping, and values outside valid ranges for ages and dates. Additional complexity arises from the need to validate domain-specific codes and conventions, including the International Classification of Diseases for Oncology (ICD-O) and the Tumor–Node–Metastasis (TNM) staging system.

In this work, we perform an exploratory analysis of cancer registry data in Brazil, process and standardize hospital oncology records, and create derived temporal features for disease trajectory analysis. Additionally, we prepare a high-quality dataset for survival prediction modeling and leverage machine learning and deep learning techniques to model patient survival.

We develop a robust preprocessing pipeline that cleans, validates, and standardizes large-scale hospital oncology records, resulting in a high-quality dataset ready for machine learning applications. Next, we perform a comprehensive evaluation of predictive models for two complementary tasks: binary vital status classification and time-to-event survival prediction. The evaluated methods include random forest, XGBoost, multilayer perceptron (MLP), AutoGluon, ensemble techniques, XGBoost Survival, and DeepSurv. The proposed models demonstrate strong predictive performance, with the best classification model achieving an ROC-AUC of approximately 0.88 and the DeepSurv survival model reaching a C-index of 0.86. Finally, we complement the predictive analysis with model interpretability techniques, including permutation importance and SHAP analyses, showing that primary tumor location, metastatic staging, patient age, and treatment-related variables are the most influential predictors of patient outcomes.

The prediction task considered in this work is performed at the beginning of the initial treatment, when the patient's diagnostic workup has been completed and the initial therapeutic strategy has been established. Therefore, the predictive models use only information available up to that clinical time point to estimate subsequent survival outcomes.

The remainder of this paper is organized as follows. Section 2 describes the data source,

[1] `https://www.gov.br/inca/pt-br/assuntos/cancer/numeros/registros`

[2] `https://www.kaggle.com/datasets/rafatrindade/onco-360/data?select=raw_inca_registro_hospitalar.parquet`

preprocessing pipeline, feature engineering, and modeling strategy. Section 3 presents exploratory findings, predictive modeling results, and their interpretation. Section 4 summarizes the main contributions, limitations, and directions for future work. The code repository used to develop the analyses can be found on GitHub[3].

2 Methodology

2.1 Feature Engineering

We implemented a PySpark-based processing pipeline, a Python API for Apache Spark, within the Databricks[4] optimized platform to refine the data. The pipeline used distributed and lazy evaluation to process the source dataset in a serverless notebook environment.

For this purpose, we used the raw dataset `raw inca registro hospitalar.parquet`[5] as a baseline, initially containing 47 variables and records corresponding to INCA's oncology hospital records. The first step consisted of standardizing the nomenclature of the 47 original columns to ensure taxonomic consistency throughout the modeling lifecycle.

To improve readability and downstream analysis, all variables were renamed using a more intuitive nomenclature. The data structure includes demographic information (age, sex, race/color, education, occupation), clinical descriptors (histological type, primary tumor site, TNM (*Tumor, Node, Metastasis*) clinical staging and pTNM pathological staging (determined after surgical and histopathological evaluation), temporal events (diagnosis, first consultation, treatment initiation, death), lifestyle factors (smoking and alcohol history), and treatment outcomes (treatment type and disease status at the end of treatment).

This initial phase of understanding the problem involved delving deeper into medical concepts to grasp the definition of each column in the dataset, in order to correctly identify the correlations between features.

Aiming to optimize the dataset dimensionality and eliminate redundancies or high missing value rates, an initial filtering was performed, resulting in the exclusion of rows and the following columns. The dataset went from 5.4 million rows to 2.9 million rows after applying the filter to retain only non-null values. These were the 19 removed variables along with their respective technical justifications:

- `probable primary location`, `screening date`, `screening year`, `clinical staging group`, and `clinical staging tnm`: Excluded due to a high index of missing values or extreme sparsity, as in the case of probable primary location, which contained more than 99% null values.
- `hospital municipality`, `hospital cnes code`, `residence municipality code`, and `birthplace`: Removed due to geographical or analytical redundancy. The state level (`hospital state` and `state of origin`) already provided the necessary spatial granularity, making the municipal or birth data dispensable for the current scope.

[3] `https://github.com/Data-Squad-ML/cancer_survival_prediction`

[4] `https://www.databricks.com/br`

[5] `https://www.kaggle.com/datasets/rafatrindade/onco-360/data?select=raw_inca_registro_hospitalar.parquet`

- `hospital treatment start year`, `first consultation year`, `diagnosis date`, and `primary location 3 digits`: Discarded in favor of other ecosystem variables that offered greater completeness, informational detail, or a complete history for temporal and topographic modeling.
- `treatment start clinic` and `first care clinic`: Deprioritized due to lower statistical relevance when compared to the predictive impact of the overall hospital structure.
- `tumor laterality`, `other clinical stagings`, and `marital status`: Evaluated as irrelevant to the established predictive goals or discarded due to a lack of theoretical specificity and the absence of structured information.
- `text`: Removed due to zero variance, meaning all records contained exactly the same value, adding no discriminatory power to the model.
- `reason no hospital treatment`: Removed due to the potential for information leakage, as some categories directly reflect the patient's outcome (e.g., death) or information only available after the prediction time point.

Furthermore, only analytical cases were retained for model development. Non-analytical cases generally correspond to patients whose diagnosis or treatment occurred outside the reporting institution, resulting in limited clinical follow-up and incomplete outcome information. Restricting the analysis to analytical cases increases the consistency of the available clinical history and improves the reliability of the survival modeling process.

The prediction time point adopted in this study corresponds to the beginning of the patient's initial treatment. Consequently, only variables that would be available at or before this time point were considered as predictors. Variables containing information generated after treatment initiation or directly reflecting the outcome were excluded to avoid information leakage.

2.2 Data Standardization and Consistency

The temporal variables of interest (`first diagnosis date`, `first consultation date`, `hospital treatment start date`, and `death date`) were parsed into a native date type and represented in ISO 8601 format (`yyyy-MM-dd`), and cutoff filters were applied to delimit the valid time interval between January 01, 1900, and December 31, 2023.

To ensure the biological and operational integrity of the data, a logical validation protocol based on temporal inequalities was established. Thereby, it was guaranteed that the sequence of events strictly obeyed the oncology treatment chronology shown in Inequality 1. No fixed ordering between first consultation and diagnosis was imposed, because diagnostic confirmation may occur after the first hospital consultation.

$$\text{Diagnosis} \leq \text{Treatment} \leq \text{Death} \tag{1}$$

Records that violated these business rules were removed. As for the null values in the `death date` column—which assist in defining the target—they were explicitly retained,

serving as a clinical indicator of right-censoring (i.e., patients alive until the end of the follow-up period). Meanwhile, the `age` variable was converted to an integer and filtered between 0 and 100 years.

The original clinical and pathological staging presented severe completeness limitations, yet it proved to be a highly important variable for the context. These consist of two acronyms, TNM and pTNM, for medical classifications: the former recorded through clinical evidence and the latter through pathological evidence. The letter "T" refers to the primary tumor size or extent, "N" refers to the involvement of nearby lymph nodes, and "M" refers to the spread to the rest of the body through metastasis.

The TNM column had 928,000 non-null records, pTNM had 561,000, and 389,000 rows contained both non-null TNM and pTNM. Given the still substantial volume and the relevance of these features to the context, it was decided to keep both columns. To analyze each component individually, each letter of each acronym was split into a distinct feature, and rows with inconsistent T, N, and M values were removed.

The validation of categorical variables was strictly based on the INCA data dictionary that accompanies the dataset.[6] Fifteen features, encompassing demographic and regional profiles, behavioral histories of smoking and alcohol consumption, diagnostic methods, treatment procedures, and clinical outcomes, had their discrete numerical values filtered and analyzed, checking if the values are consistent and within the expected limits for each case.

For complex textual variables, filters via regular expressions (*regex*) were applied. The `main occupation` variable was validated to ensure a strict non-zero numerical format[7]. Codes were subsequently validated against the applicable version of the Brazilian Classification of Occupations (CBO). Values with a valid numerical shape but absent from the reference domain were treated as invalid or unknown.

Concurrently, clinical variables from the International Classification of Diseases for Oncology (ICD-O) were filtered to guarantee morphological[8] and topographic[9] structural compliance. Any record failing to comply with these standardized domain constraints was excluded from the dataset.

From the validated topography and morphology fields, string-type records were also handled. The `detailed primary location` variable was decomposed into a main category (`primary location category`) and a subcategory (`primary location subcategory`). Analogously, the `tumor histological type` column was reconfigured to isolate the base numerical code and extract the `tumor histological behavior` (the digit following the slash).

Finally, the feature engineering stage structured five new variables focused on the outcome and temporal latency metrics:

1. `vital status` **(classification target):** Binarily encoded (0 for deceased, 1 for alive), based on whether `death date` was filled or on the clinical discharge code. Eighty-two

[6] `https://raw.githubusercontent.com/rafa-trindade/oncoped-360/main/docs/dicionario_inca_registro_hospitalar.pdf`

[7] Validated using the regex pattern: `^[1-9]\d*$`

[8] Pattern: `^(8|9)\d{3}/[012369]$`

[9] Pattern: `^C(0[0-9]|[1-7][0-9]|80)\.[0-9]$`

inconsistent records that indicated death but lacked a recorded date were removed. The volume of the target for deceased cases was thus settled at 12.5% of the dataset.

2. `total disease time`**:** Measures the days of survival or follow-up. For alive patients, the interval up to the right-censoring date (December 31, 2023) was calculated; for deceased cases, the date of death was used.

3. `time until consultation` **and** `time until treatment`**:** These variables measure the intervals, in days, from diagnosis to the corresponding clinical events. Since the prediction time point adopted in this study corresponds to the beginning of the initial treatment, both variables are fully available at that moment and were therefore included as predictive features.

4. `time until death`**:** This variable represents the interval, in days, between diagnosis and death. It was used exclusively to define the survival outcome, representing the interval from diagnosis to death or to the administrative censoring date, December 31, 2023. Because this variable directly determines the survival outcome, it was not included among the predictive features, thereby avoiding information leakage.

It is worth mentioning that the available database does not provide complete longitudinal follow-up information for every patient. Consequently, records without a documented date of death were treated as censored observations or as surviving cases according to the adopted analysis strategy. Although this assumption is commonly required when working with secondary healthcare databases, some patients may have died outside the observation system, potentially leading to an overestimation of predictive performance. This limitation reflects one of the main challenges of real-world clinical datasets, in which complete follow-up information is often unavailable.

Another limitation inherent to the available data was that patient identifiers were not provided in a form that allowed explicit verification of whether multiple records from the same individual were present across the training and testing sets. Since this information was not available in the source registry, no correction could be applied at the modeling stage, and although no evidence suggests that this substantially affected the reported results, this constraint should be considered when interpreting model performance.

Furthermore, after consolidating these new metrics, all original date columns and the status of the disease at the end of treatment were removed to avoid redundancy. With this, we were able to perform a thorough diagnostic investigation that guaranteed data quality prior to the modeling itself. The use of figures, such as the example in Figure 1, aids in the rapid visualization of inconsistencies, anomalies, skewed distributions, or patterns, even if the models used are capable of perceiving much more complex patterns.

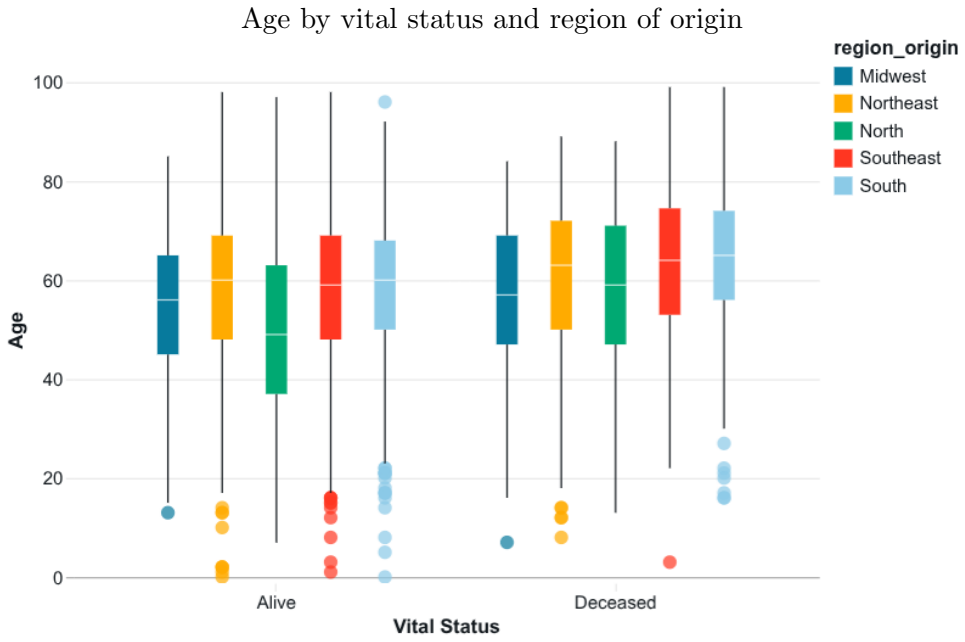

Figure 1: Example of a box plot for exploratory data analysis and target validation.

In Figure 1, we can see that the age distribution varies according to vital status (deceased or alive) and region, with slightly higher median ages for the deceased group compared to the alive group. Furthermore, it is notable that all regions follow this same pattern, which is consistent with the expected association between age and mortality across distinct geographical locations.

To complement this step, more exploratory data analysis was carried out in order to better understand the overall structure and behavior of the dataset, focused on identifying how the data is distributed across the available samples.

As illustrated in the subsequent figures (2, 3, 4), some variables exhibit mild to moderate concentrations in specific ranges, which is expected in real-world clinical datasets. Despite these concentrations, the dataset maintains a broad and diverse distribution overall, suggesting that it contains sufficient variability to support model training and generalization.

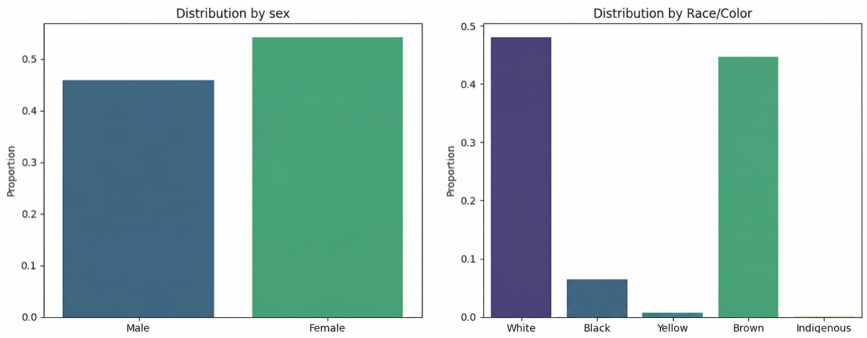

Figure 2: Distribution of sex and race/color categories in the final analytic cohort.

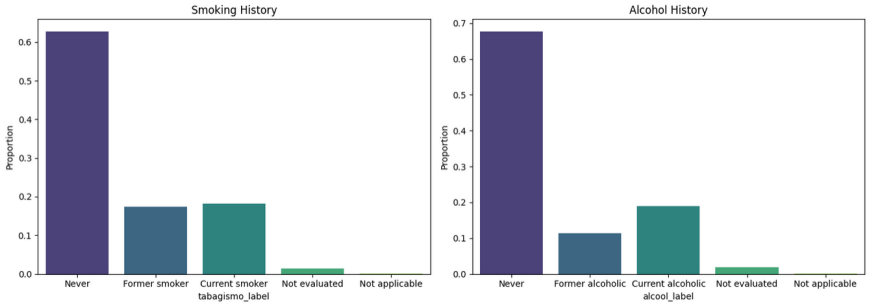

Figure 3: Distribution of smoking and alcohol-history categories in the final analytic cohort.

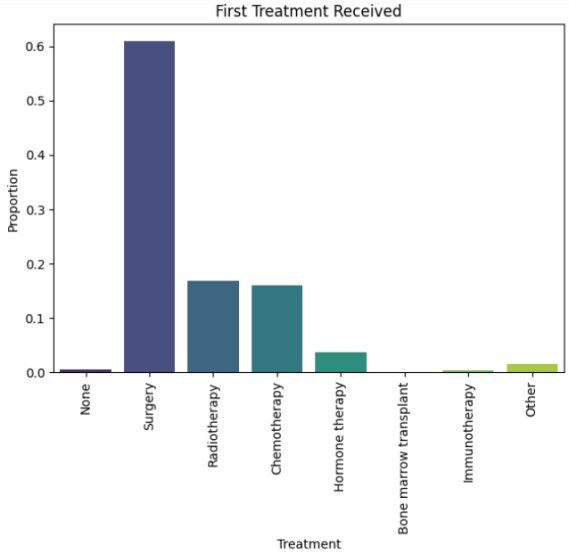

Figure 4: Distribution of the first treatment received at the reporting hospital.

Ultimately, as a result of the preprocessing pipeline we obtained a final cohort of 73,459

patients, substantially reduced when compared with the original database, which contained more than 5.4 million records. As previously described in this section, this reduction was primarily driven by the exclusion of records with missing values in essential variables, inconsistent dates, implausible values, duplicated information, and other data quality issues commonly found in large-scale clinical databases. The purpose of this process was to obtain a reliable and internally consistent cohort suitable for machine learning. In predictive modeling, data quality and representativeness are generally more important than the absolute sample size.

Therefore, the resulting cohort should be interpreted as a high-quality subset of the original population, containing sufficient clinical information to support robust model development and evaluation. Although considerably smaller than the original database, the final cohort remains sufficiently large for machine learning applications while providing substantially higher data completeness and consistency.

2.3 Encoding

The final processing step consisted of the structured encoding of the attribute ecosystem to adapt them to the requirements of the classification and survival models. Since there were no longer any null values to be handled, the goal was to transform the features into numerical formats, understanding the circumstance of each variable to analyze its matching with each encoding method.

The primary point of attention for selecting the encoding method was heavily based on the features' cardinality and the need for an inherent order among their values. This is because One-Hot Encoding can represent categorical features very well, but should be avoided for high-cardinality columns to prevent creating too many sparse columns. On the other hand, Label Encoding, for instance, handles high cardinality well, but ends up establishing an order and a uniform spacing among the encoded numbers that might not be intentional, as it may not semantically reflect the actual differences between the original labels.

The use of raw alphanumeric codes, such as pure ICD-O or CBO records, was avoided, as these would introduce an artificial arithmetic ordering without any clinical correspondence. Thus, the purely quantitative variables of time and age were kept in their original continuous scale (*double*). Meanwhile, variables classified as binary underwent a uniform mapping, converting the native intervals $[1, 2]$ to the boolean format $[0, 1]$, preserving `vital status` as the target variable of the classification study [15].

For the attributes of a categorical nature, the encoding strategy was segmented according to the clinical properties and cardinality of each field. Legitimate ordinal variables that express a real pathological or hierarchical progression—such as the tumor staging components (`t tnm`, `n tnm`, `m tnm`, and their pathological equivalents), educational level, and tumor histological behavior—were preserved via label encoding so that the model can capture the gradation of severity.

Conversely, nominal variables with low cardinality were subjected to one-hot encoding (OHE). As an optimization criterion for this process, spatial data (`state of origin` and `hospital state`) were previously aggregated into geographical macro-regions, while the

codes for `primary location category` with a frequency lower than 1% were consolidated under the single label "others". High-cardinality fields were mitigated using target encoding techniques for the micro-topographical data and gap encoding for the `main occupation` variable.

The analytic split was created before any data-dependent preprocessing. All scaling, rare-category grouping, one-hot encoding, gap encoding, and target encoding operations were fitted exclusively on the training data. During cross-validation, these operations were refitted within each training fold. Target encoding was computed using out-of-fold estimates in the training data and was subsequently applied to the validation and test sets without using their outcomes.

2.4 Master table

After completing the data cleaning and feature processing steps described in the previous sections, the final dataset was obtained. The reduction in the total number of samples is mainly a consequence of the filtering procedures applied during preprocessing, particularly the removal of records with missing, incomplete, or inconsistent information. These steps were necessary to ensure data quality and reliability for the subsequent modeling phase.

The final predictor set consisted of demographic, clinical, pathological, and treatment-related variables available at the beginning of the initial treatment. The retained numerical and ordinal variables were: `age`, `time until consultation`, `time until treatment`, `sex`, `family history of cancer`, `multiple primary tumors`, `education level`, `clinical TNM staging (T, N, M)`, `pathological TNM staging (pT, pN, pM)`, `histological tumor behavior`, indicators of missing information for smoking and alcohol history, `tumor histological type`, `primary tumor subsite`, `primary tumor site`, and the grouped primary occupation variable.

Categorical variables were encoded using one-hot encoding, including `race/color`, `region of patient's origin`, `hospital region`, `referral source`, `relevant diagnostic examinations`, `previous diagnosis/treatment`, `most important diagnostic basis`, `microscopic diagnostic basis`, `first hospital treatment`, `clinical smoking history`, and `clinical alcohol consumption history`.

Furthermore, the dataset was divided into training and test sets, preparing the data for the machine learning models. At this stage, a significant class imbalance in the target variable was identified, as shown in Figure 5. This is a common characteristic in medical datasets and can negatively impact model performance by biasing predictions toward the majority class.

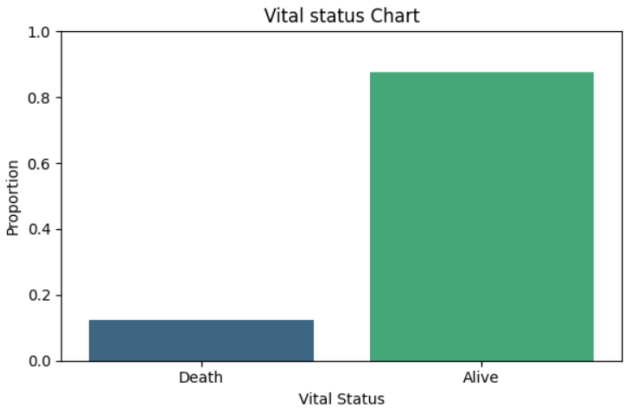

Figure 5: Vital status proportion

To mitigate this issue, a class weighting strategy was adopted using a balanced weighting approach inspired by Scikit-learn's `class_weight='balanced'` method. In this approach, class weights are computed based on the inverse frequency of each class, ensuring that minority class samples contribute more strongly during training. This helps reduce bias and improves the model's ability to learn patterns from underrepresented cases.

The final dataset consisted of 58,767 training samples and 14,692 testing samples. Class weights were computed only from the training labels and were used during model fitting; no weights were estimated from the test set.

2.5 Binary Classification Models

After the data had been properly structured and preprocessed, predictive modeling using machine learning was performed to classify patients' vital status into two categories (alive or deceased). Classification models aim to learn patterns from labeled data in order to predict the class to which a new observation belongs. In this study, the problem consists of determining a patient's vital status using the available clinical and sociodemographic variables as input.

Three supervised classification models were evaluated: random forest, XGBoost, and multilayer perceptron (MLP). In addition to these individual models, strategies based on *Automated machine learning* (AutoML), using the AutoGluon library, and ensemble techniques, through the soft voting and hard voting methods, were also investigated. The objective was to combine the predictions of the individual classifiers and achieve improved predictive performance.

All predictive models were trained under the assumption that predictions are generated immediately after the initial treatment has been defined. For information on configuring models, libraries, and hyperparameters, detailed explanations are provided in the repository[10].

[10] `https://github.com/Data-Squad-ML/cancer_survival_prediction`

2.5.1 Random Forest

For the first model, random forest [2], the adopted hyperparameter optimization strategy was Grid Search, combined with *K-Fold* cross-validation with $K = 5$ [26]. This approach partitions the training set into five mutually exclusive subsets, using four subsets for model fitting and one for validation iteratively, ensuring that the estimated performance is robust and less susceptible to random sampling variations. The primary target metric used to guide the selection of the best hyperparameters during cross-validation was the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which measures how well the model is able to distinguish between the two classes [11].

Additionally, due to the inherently imbalanced nature of oncological medical data, the training of all algorithms systematically employed class frequency-based sample weights (*sample weights*), mitigating the learning bias toward the dominant class.

Random forest [2] is a supervised learning algorithm based on the concepts of ensemble learning and bootstrap aggregation (bagging). The method operates by constructing a large collection of independent decision trees during the training phase. Each tree is trained on a random bootstrap sample of the data, while node splits consider only a random subset of the available features. This mechanism introduces diversity into the ensemble, substantially reducing the variance of the final model and its susceptibility to overfitting compared with a single deep decision tree.

The implementation followed a procedural structure. First, the data are loaded and validated, and the categorical and numerical columns are separated and mapped into data frames and series. A GridSearchCV procedure with *K-Fold* cross-validation [26] evaluates combinations of hyperparameters such as the total number of estimators (trees), the maximum tree depth, and the minimum split and leaf criteria (*min_samples_split* and *min_samples_leaf*). After determining the optimal hyperparameters based on the AUC metric, the model is retrained using the complete training dataset. Finally, the script performs a Permutation Importance analysis by iteratively shuffling the columns of the test set to measure the resulting decrease in model performance, thereby ranking the variables that are most relevant for clinical prediction.

This model did not require computationally intensive training to achieve satisfactory results. Increasing the hyperparameter search space or evaluating additional optimization criteria did not yield significant performance improvements, but instead increased the risk of overfitting. Therefore, the experiments were executed on the CPU, which adequately met the computational requirements. A GPU implementation would have required a modern graphics card and code adaptations, as the library does not provide native GPU support for this implementation.

2.5.2 Extreme Gradient Boosting

XGBoost [4] is a widely used implementation of gradient-boosted decision trees. Unlike random forest, which trains decision trees independently and in parallel, the boosting paradigm constructs the model sequentially. Each new decision tree is specifically designed to predict and correct the residual errors produced by the ensemble of previously trained trees,

minimizing a global loss function through the gradient descent optimization process. Additionally, XGBoost incorporates both L1 and L2 regularization penalties on tree leaves, making it highly robust to noise in the data.

A notable structural characteristic of this implementation is its hardware adaptability. The script includes an environment detection routine that attempts to instantiate GPU-based histogram tree construction methods (`gpu_hist` or `cuda_hist`). If GPU acceleration is unavailable or driver incompatibilities are detected, the implementation automatically and transparently falls back to CPU execution. The entire hyperparameter optimization process—which explores combinations of learning rate, subsampling ratio, maximum tree depth, and minimum child weight (*min_child_weight*)—is performed using the optimized `GridSearchCV` class [26]. The script concludes by generating a comprehensive plain-text report containing a wide range of descriptive performance metrics, summarizing not only the model's discriminative ability but also its probability-based threshold performance.

2.5.3 Artificial Neural Network

The third implemented model consists of an Artificial Neural Network based on the classical *Multilayer Perceptron* (MLP) architecture [13]. From the perspective of deep learning, an MLP acts as a universal function approximator, consisting of an input layer that maps the input features, a sequence of hidden layers responsible for extracting hierarchical representations, and an output layer that estimates the probability of membership in the vital status classes. Each connection is associated with a synaptic weight that is iteratively updated through the backpropagation algorithm.

Unlike tree-based methods, which partition the feature space using orthogonal splits, neural networks critically depend on the geometric scale of the input data. Because MLP optimization is sensitive to feature scale, continuous input variables were standardized using `StandardScaler`. The scaler was fitted only on each training fold and then applied to the corresponding validation or test data.

Structurally, the implementation was entirely developed in `PyTorch`, following an object-oriented design and optimized to maximize GPU performance. The model class is dynamically configurable and instantiated using blocks composed of linear projection layers, interleaved with modern learning stabilization techniques. Specifically, Batch Normalization is applied to accelerate convergence and mitigate the vanishing gradient problem, followed by nonlinear activation functions (such as ReLU or Tanh) and stochastic neuron masking through Dropout, which prevents spurious dependencies and improves network regularization.

The training loop optimizes the Cross-Entropy Loss function using mini-batches and the Adam optimizer. Class weights are also incorporated into the `PyTorch` loss function to account for class imbalance. The Grid Search and *K-Fold* cross-validation procedures were manually implemented through nested loops to explore combinations of architectural hyperparameters, including hidden layer dimensions, dropout rate, batch size, number of training epochs, and learning rate.

2.5.4 Automated machine learning

In addition to the individually evaluated classification models, an *Automated machine learning* (AutoML) approach was also employed using the *AutoGluon Tabular* library [1, 10]. Unlike traditional machine learning algorithms, AutoGluon does not correspond to a single predictive model but rather to a framework that automates the training, selection, and combination of multiple learning algorithms. The framework automatically identifies the input variable types, applies algorithm-specific preprocessing, performs hyperparameter optimization, and compares model performance according to a predefined evaluation metric.

In this study, AutoGluon was configured to solve a binary classification problem using ROC-AUC as the primary evaluation metric during training. The same input features employed by the other models were used, ensuring a fair comparison among the different approaches. The set of available algorithms included tree-based models such as random forest and extra trees, gradient boosting algorithms including lightGBM, XGBoost, and catBoost, as well as neural networks implemented in PyTorch.

One of AutoGluon's main advantages is its automatic construction of multi-level stacking models and weighted ensembles. In the stacking procedure, different base models are initially trained on the original dataset, and their predictions are subsequently used as input features for higher-level models. This process enables subsequent models to learn the complementary error patterns produced by lower-level predictors. At the end of training, AutoGluon automatically builds a *Weighted Ensemble*, which combines the predictions of the best-performing models or intermediate ensembles using weights learned during training, aiming to maximize the selected evaluation metric. Consequently, the final model represents an optimized combination of different learning algorithms.

Training was performed using the `high_quality` preset, which prioritizes predictive performance over training time. In addition, five-fold cross-validation and two stacking levels were employed, allowing the automatic construction of hierarchical ensemble models.

Upon completion of training, AutoGluon generates a leaderboard containing all trained models and their respective performances, automatically selecting the best-performing model according to the ROC-AUC metric. The final evaluation was conducted on an independent test set using the predicted probabilities for the positive class to compute ROC-AUC, together with additional performance metrics provided by the library. This approach enables the automatic exploration of multiple algorithms, hyperparameter configurations, and ensemble strategies, substantially reducing the manual effort required for model development and optimization.

2.5.5 Ensemble

Ensemble methods combine the predictions produced by multiple machine learning models with the objective of achieving superior performance compared with individual classifiers. The underlying motivation is that different models may commit different errors on the same dataset; therefore, aggregating their predictions tends to reduce variance, increase robustness, and improve the classifier's generalization capability [9, 36].

In addition to the ensemble strategy automatically generated by AutoGluon, indepen-

dent hard voting and soft voting approaches were implemented using the previously trained random forest, XGBoost, and MLP (deep learning) models. Since all models were evaluated on exactly the same test samples, their predictions could be combined directly without requiring additional training.

The objective of this stage was to evaluate whether explicitly combining the best-performing classifiers developed in this work could improve predictive performance relative to the individual models. Unlike AutoGluon, which constructs ensembles during the training and model selection process, the proposed ensemble methods were built *a posteriori*, using only the predictions generated by the final trained models. This approach enables a direct comparison between ensemble strategies while providing greater control over the participating algorithms and the aggregation rule.

In soft voting, class-probability estimates from the component models were combined using an equal-weight average. Before aggregation, probability calibration was evaluated on validation data because models with differently calibrated outputs may contribute disproportionately to a simple average. Ensemble weights and any calibration transformations were selected without using the held-out test set. The final predicted class corresponds to the one with the highest aggregated probability. Since the ensemble preserves a continuous probability score, ranking-based metrics such as ROC-AUC, Precision-Recall AUC, KS statistic, Gain, Lift, as well as calibration metrics including Brier Score and Log-Loss, can be computed.

Conversely, hard voting combines only the predicted class labels from each individual model. Each classifier first predicts its most probable class, after which the final prediction is obtained through majority voting. Because the ensemble consists of three models, tie situations cannot occur. Unlike soft voting, this strategy does not generate aggregated probabilities but only the final predicted class. Because hard voting produces only class labels rather than meaningful continuous scores or calibrated probabilities, ranking- and probability-based metrics were not reported for this ensemble. Therefore, only class-based evaluation metrics, such as accuracy, precision, F1-score, Matthews Correlation Coefficient (MCC), and the confusion matrix, are reported.

Therefore, the combined use of soft voting and hard voting makes it possible to evaluate different strategies for combining individual models. While hard voting emphasizes consensus among classifiers, soft voting additionally incorporates the confidence associated with each model's predictions.

2.5.6 Performance Evaluation Metrics

To enable a comprehensive and consistent descriptive analysis of the trained models, a centralized and standardized performance evaluation module was implemented across all developed scripts. These metrics assess model performance through three complementary evaluation dimensions, providing a detailed characterization of the inference process on the test set:

- **Discrimination Metrics:** These evaluate the classifier's ability to consistently rank observations according to their likelihood of belonging to each class. They include

the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which measures the trade-off between the true positive rate and the false positive rate across different decision thresholds; the Area Under the Precision-Recall Curve (PR-AUC), particularly important for imbalanced classification problems; and the Gini Index, which is directly derived from the ROC-AUC [11]. ROC-AUC and PR-AUC range from 0 to 1, with higher values being better. The Gini Index ranges from -1 to 1 (commonly from 0 to 1 in practical applications), with higher values being better.

- **Classification Metrics:** These quantify the classifier's performance at a specific decision threshold (typically the default probability threshold). They include Overall Accuracy, Precision (Positive Predictive Value), Recall (Sensitivity), the F1-score (the harmonic mean of Precision and Recall) [31], and the Matthews Correlation Coefficient (MCC) [29], which is particularly robust in the presence of class imbalance [5]. Accuracy, Precision, Recall, and F1-score range from 0 to 1, with higher values being better. MCC ranges from -1 to 1, where 1 indicates perfect prediction, 0 corresponds to random prediction, and -1 indicates complete disagreement. Therefore, higher values are better.

- **Calibration Metrics:** These assess the reliability of the predicted probability estimates. The *Brier Score* measures the mean squared difference between predicted probabilities and binary outcomes and ranges from 0 to 1, with lower values being better [3]. Log-loss (logarithmic loss), in turn, heavily penalizes predictions that are both highly confident and incorrect [17]. Both metrics range from 0 to $+\infty$, with the optimal value equal to 0. Therefore, lower values are better.

Additionally, for binary classification tasks, the developed scripts generate reports containing operational decision-support metrics, most notably the *Kolmogorov-Smirnov* (KS) statistic. This is the maximum difference between the cumulative score distributions of the positive and negative classes, equivalently the maximum value of sensitivity minus false-positive rate across decision thresholds. A decile table provides only a discretized approximation of this maximum. The statistic ranges from 0 to 1, with higher values being better. The reports also include *Lift* tables, which quantify the improvement achieved relative to random selection. Lift ranges from 0 to $+\infty$, with a value of 1 indicating performance equivalent to random selection and higher values being better. This standardized evaluation framework ensures that all implementations produce directly comparable performance reports across both machine learning and deep learning approaches.

2.6 Survival Models

Expanding upon traditional binary classification analysis, survival analyses (*Time-to-Event Analysis*) were conducted. While conventional binary classification is limited to predicting the occurrence or non-occurrence of an outcome within a fixed time horizon (such as the `vital status` mapped at a static point in time), survival analysis explicitly models the time elapsed until the event of interest occurs. This paradigm shift is advantageous for mitigating selection biases and avoiding information loss due to censoring (predominantly

right-censoring) [6]. Censoring occurs when the event of interest is not observed in certain individuals during the follow-up period, whether due to loss to follow-up, patient withdrawal from the study, or the chronological end of data collection without the outcome occurring [25].

To relax the linearity of the conventional Cox risk function [7], two nonlinear Cox-type models were investigated: *XGBoost Survival* and *DeepSurv* (neural network).

2.6.1 XGBoost Survival

The *XGBoost Survival* model [4] is an approach based on gradient tree boosting configured with the Cox proportional hazards objective function (`objective = 'survival:cox'`) and optimized using the negative Cox partial likelihood loss function (`eval_metric = 'cox-nloglik'`). To enable mathematical processing in tree algorithms, the bivariate survival response variable was converted into a one-dimensional vector of labels, where censored instances assume negative time values and events assume positive values. The script implements an automation mechanism that sequentially tests GPU-accelerated histogram-based tree construction methods (`cuda_hist` and `gpu_hist`), ensuring computational efficiency. Furthermore, an early stopping strategy was incorporated, configured to halt training after 75 consecutive rounds with no improvement in validation loss, actively mitigating the risk of overfitting based on a 15% fraction of training samples set aside exclusively for this purpose.

Inverse Probability of Censoring Weighting (IPCW) was used for censoring-aware evaluation, including Uno's C-index, time-dependent AUC, and the Brier score. Within cross-validation, the censoring distribution was estimated exclusively from each training fold and applied to its validation fold. Final test metrics used censoring weights estimated from the full training set. The XGBoost Cox model itself was trained by minimizing the negative Cox partial likelihood without conventional class weighting.

Cross-validation was executed using a *StratifiedKFold* strategy with $K = 5$, where stratification is based on the binary event occurrence indicator. To mitigate statistical instabilities across validation folds, a check was implemented to enforce a minimum threshold of 5 events per *fold*. The Grid Search was guided by the optimization of the Uno Concordance Index (*Uno C-index* via IPCW). The hyperparameter space swept encompassed the number of estimators (`n_estimators`), maximum depth (`max_depth`), learning rate (`learning_rate`), subsampling fractions (`subsample` and `colsample_bytree`), as well as L1 (`reg_alpha`) and L2 (`reg_lambda`) regularization parameters, structural complexity penalties (`gamma`), and minimum child node weight (`min_child_weight`).

With `objective="survival:cox"`, XGBoost produces relative-risk scores rather than calibrated survival probabilities. Therefore, model evaluation focused on discrimination-based metrics, including the c-index, uno's c-index, time-dependent AUC, kaplan-meier risk stratification, decision curve analysis, and feature importance analyses. Kaplan-Meier curves were generated by stratifying patients into risk groups according to the predicted relative-risk scores, allowing visual assessment of survival separation between groups. Additionally, permutation importance was computed.

2.6.2 DeepSurv

DeepSurv [24] is a feedforward artificial neural network (or *Multi-Layer Perceptron* - MLP) designed specifically to extend the classical Cox proportional hazards model, allowing for the learning of non-linear relationships between covariates and the individual's log-hazard. The data preparation workflow began with the standardization of numerical variables using `StandardScaler`, ensuring that the features had zero mean and unit variance, a requirement for the convergence of gradient-based algorithms.

The network architecture was structured modularly using the PyTorch library with support for GPU processing (`CUDA`). Each hidden layer consisted of linear transformations, followed by normalization layers and regularization mechanisms via *Dropout* to prevent overfitting. The model was parameterized to evaluate both conventional activation functions (`ReLU`) and self-normalizing functions (`SELU`). In the arrangement containing the `SELU` activation, the *BatchNorm* layers were omitted due to structural incompatibility, and the conventional *Dropout* was replaced by *AlphaDropout* to preserve the mean and variance properties of the neurons, also applying a `Kaiming normal` weight initialization. The output layer of the MLP consists of a single linear neuron without an activation function, whose predicted scalar value represents the patient's relative log-hazard score.

The training of the neural network was based on minimizing the Cox negative partial log-likelihood loss function (*Negative Partial Log-Likelihood*), computed under the Breslow approximation for handling tied times. The loss function was extended to natively incorporate sample weighting by inverse probability of censoring weighting (IPCW) or by class weights, aligning the learning with the weights calculated in the Kaplan-Meier estimator of censoring. The optimization process used the Adam algorithm with L2 penalization via weight decay, processing the data in minibatches of size 512 with stochastic reshuffling at each epoch. To stabilize the learning dynamics, an adaptive learning rate decay was implemented using the `ReduceLROnPlateau` scheduler, which halves the initialization parameter if the loss stagnates for 10 consecutive epochs. In addition, a gradient clipping technique configured with a maximum norm of 1.0 was applied to mitigate the exploding gradient problem.

Hyperparameter optimization was conducted through a manual Grid Search integrated with a *StratifiedKFold* cross-validation strategy with $K = 5$, using the binary event occurrence indicator as the stratification criterion. The performance of each combination in the Grid Search was guided by the maximization of the *Uno C-index*. The search space encompassed different hidden layer topologies (`hidden_sizes` $\in \{[128, 64], [256, 128], [256, 128, 64]\}$), dropout rates ($\in \{0.1, 0.3\}$), initial learning rates ($lr \in \{10^{-3}, 3 \times 10^{-4}\}$), and activation functions. Following the convergence of the best set of hyperparameters and the retraining of the model on the consolidated dataset, the predicted log-hazard scores for the test set were submitted to a customized formulation of the non-parametric Breslow estimator to reconstruct the baseline cumulative hazard function.

2.6.3 Performance Evaluation Metrics

The evaluation of predictive models in survival analysis differs from binary classification due to the need to incorporate the temporal dimension and handle censored data [35]. The central goal of validating the algorithms is to verify how accurately they estimate the **Survival Function** $S(t)$, which defines the probability that an individual will not experience the event of interest up to a specific time t, expressed mathematically as:

$$S(t) = P(T > t)$$

To measure the quality of these estimates from different perspectives, the performance of the models was mapped using metrics divided into three main pillars: discrimination, calibration, and clinical utility.

Prognostic Discrimination Metrics

Discrimination metrics evaluate the model's ability to correctly distinguish between high- and low-risk individuals, meaning whether the patient who experienced the event earlier received a higher risk score than the one who survived longer.

- **Harrell's and Uno's Concordance Index (C-index) (IPCW):** The C-index evaluates the proportion of patient pairs whose prognoses were correctly ranked by the model. In addition to Harrell's classical formulation [16], the *Uno C-index* was used, which corrects the estimation bias in long time horizons by incorporating weights based on Inverse Probability of Censoring Weighting (IPCW) [35]. The metric ranges from 0 to 1, where 0.5 corresponds to random discrimination, 1.0 indicates perfect concordance, and values below 0.5 indicate performance worse than random. Therefore, higher values are better.

- **Cumulative Dynamic Area Under the ROC Curve ($AUC(t)$):** Unlike the static ROC curve, the dynamic AUC calculates the model's discriminative capacity at specific time points (t) during follow-up [18]. It quantifies how well the model separates individuals who experienced the event by time t (cases) from those who survived beyond t (controls). Overall performance is summarized by the integrated mean dynamic AUC. The metric ranges from 0 to 1, where 0.5 represents random discrimination and 1.0 indicates perfect discrimination. Thus, higher values are better.

- **Royston-Sauerbrei D Statistic:** Royston's D coefficient (`D_royston`) measures the model's discriminative power directly on the log-hazard scale [33]. A value of $D > 1$ indicates a reasonable prognostic separation between risk profiles, while values greater than 2 denote excellent separation capability. The indicator is accompanied by R^2_{Royston}, a measure derived from prognostic separation. It should not be interpreted as the literal proportion of variance in observed survival times explained by the model. The statistic has no fixed upper bound, with a theoretical range from 0 to $+\infty$, where larger values indicate stronger prognostic separation. Therefore, higher values are better.

Calibration Metrics and Temporal Accuracy

While discrimination focuses on ranking risks, calibration evaluates whether the quantitative probability predicted by the model aligns with the actual event rate observed in the study population.

- **Time-Dependent Brier Score and Integrated Brier Score (IBS):** The *Brier Score* evaluates the mean squared error between the survival probability estimated by the model and the patient's actual status at time t while accounting for right-censoring through IPCW. The *Integrated Brier Score* consolidates this metric by integrating the errors over the entire time horizon [14]. The metric ranges from 0 to 1, with 0 representing perfect prediction and larger values indicating increasing prediction error. Consequently, lower values are better.

- **Local Calibration by Risk Decile (E/O Ratio):** Adopting logic analogous to classification gain tables, patients in the test set were divided into deciles based on their risk scores. For the median reference time, the ratio between the events expected by the model (E) and the events actually observed (O) via empirical curves was calculated. An E/O ratio close to 1.0 reflects perfect calibration [8], whereas values greater than 1.0 indicate risk overestimation and values less than 1.0 indicate risk underestimation. The ratio theoretically ranges from 0 to $+\infty$, with the optimal value equal to 1.0. Therefore, values closer to 1.0 are better.

Clinical Stratification

This group of metrics evaluates the model's potential for practical translation, determining whether its predictions can guide medical interventions and safely separate management groups.

- **Kaplan-Meier Curves by Risk Groups and Log-Rank Test:** The linear risk scores of the test set were stratified into tertiles to isolate low-, intermediate-, and high-risk clinical groups. For each stratum, the empirical Kaplan-Meier survival curve was estimated [23]. The statistical divergence between the curves of the extreme groups (low versus high risk) was validated using the log-rank test (Mantel-Cox) [28], evaluating whether the model indeed isolates subpopulations with significantly distinct prognoses. The associated log-rank statistic ranges from 0 to $+\infty$, where larger values indicate stronger separation between survival curves, while the corresponding p-value ranges from 0 to 1, with smaller values (typically $p < 0.05$) indicating stronger statistical evidence of separation.

- **Kolmogorov-Smirnov (KS) Survival Metric:** Adapted for the context of censored data, the KS survival statistic evaluates the model's separation capability by measuring the maximum cumulative vertical distance between the empirical distributions of the risk scores of individuals who experienced the event versus those who remained censored across the risk deciles [12]. The statistic ranges from 0 to 1, where 0 indicates no separation and 1 represents perfect separation. Thus, higher values are better.

3 Results and Discussion

3.1 Binary classification models

All models presented good performance on the test set, as shown in Tables 1 and 2, with ROC-AUC ranging approximately between 0.85 and 0.88, indicating a consistent discrimination capacity between outcomes. The ensemble soft voting model presented the highest PR-AUC on the test set (0.9794), while AutoGluon presented the best KS (0.5848) among the models, suggesting a good capacity for class separation across the probability distribution. AutoGluon also presented the lowest Brier Score (0.0798), closely followed by random forest (0.0848), indicating better probabilistic calibration compared to the other evaluated models.

Model	**ROC-AUC**	**PR-AUC**	**KS**	**F1-score**	**MCC**	**Brier**	**Log-Loss**
Random Forest	0.8747	0.9792	0.5796	0.9359	0.4161	0.0848	0.2804
XGBoost	0.8741	0.9788	0.5840	0.8756	0.4318	0.1359	0.4154
MLP	0.8570	0.9751	0.5453	0.8579	0.3950	0.1472	0.4408
AutoGluon	0.8722	0.9782	**0.5848**	**0.9406**	0.3608	**0.0798**	**0.2639**
Ensemble Hard	–	–	–	0.8928	0.4330	–	–
Ensemble Soft	**0.8758**	**0.9794**	0.5803	0.8993	**0.4441**	0.1116	0.3474

Table 1: Comparison of vital status classification models on the test set

"–" indicates that the metric was not reported. For the hard voting ensemble, probability-based metrics (ROC-AUC, PR-AUC, KS, Brier Score, and Log-Loss) are not applicable because the method outputs only the final class prediction through majority voting.

Model	Accuracy	Precision	Recall	Balanced Accuracy
Random Forest	0.8857	0.9190	0.9534	0.6820
XGBoost	0.8002	**0.9618**	0.8037	**0.7897**
MLP	0.7750	0.9591	0.7761	0.7718
AutoGluon (RandomForest_BAG_L1)	**0.8918**	0.9017	**0.9836**	0.6154
Ensemble Hard Voting	0.8236	0.9533	0.8396	0.7754
Ensemble Soft Voting	0.8330	0.9527	0.8516	0.7774

Table 2: Comparison of classification metrics on the test set

XGBoost showed competitive performance, with an ROC-AUC of 0.8741 and an F1-score of 0.8756 on the test set. Although inferior to AutoGluon and Random Forest in terms of F1-score, XGBoost maintained consistent results in the discrimination metrics and presented relatively stable behavior between training and testing.

Observing the results of the feature importance analysis, through the *Permutation Importance* technique for these models, we can see a high concordance between the two, indicating that both captured similar patterns present in the data, as shown in Figure 6.

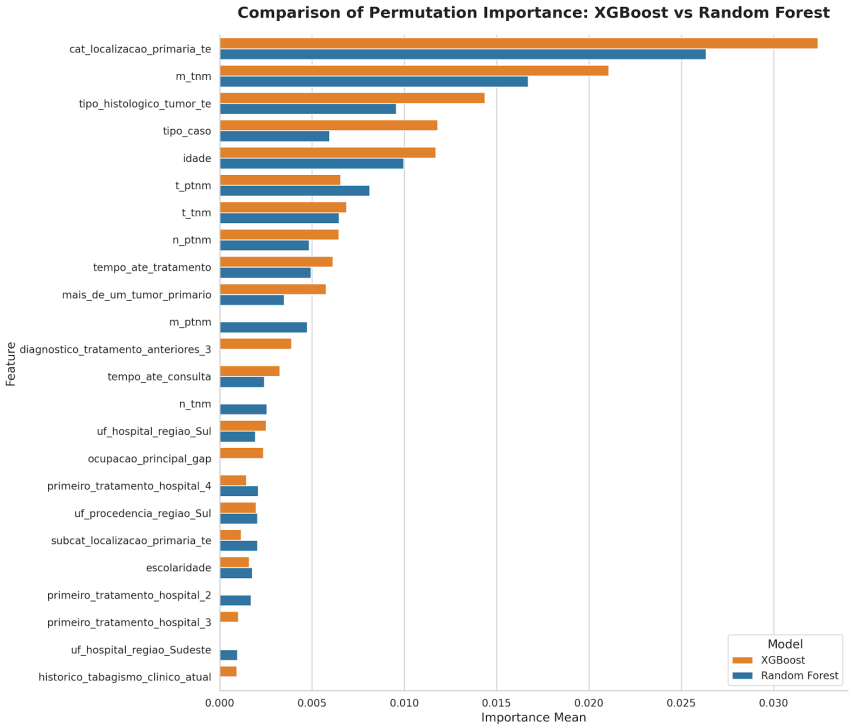

Figure 6: Random Forest and XGBoost's permutation importance comparison

In both cases, the primary tumor location category variable was identified as the feature of greatest importance, followed by the metastasis staging. These results suggest that this information plays a central role in predicting the patients' vital status.

Although there are slight differences in the feature ranking, the set of attributes considered most relevant remained practically the same between random forest and XGBoost, which increases the reliability of the interpretation of the results, as it reduces the likelihood that the observed relevance is a consequence of the particularities of a single algorithm.

Furthermore, regarding the results of the MLP neural network, it obtained an ROC-AUC of 0.8570 and an F1-score of 0.8579, demonstrating inferior performance to the tree-based models, as well as a more pronounced loss of performance relative to the training set, suggesting less generalization capability. It is worth noting that even though its performance was slightly inferior compared to the others, it still showed great performance.

The results presented in Table 3 allow for the evaluation of the models' generalization capability by comparing the performance obtained on the training and test sets. It was observed that all models showed a reduction in performance on the test set, which is expected in supervised learning tasks. However, random forest presented the largest gap between training and testing, with a reduction of 0.1241 in ROC-AUC and 0.0572 in F1-score, in addition to an increase of 0.0524 in the *Brier Score*. This behavior indicates a higher degree of overfitting, also evidenced by the practically perfect performance during training (ROC-AUC of 0.9988), which was not maintained in the test.

Furthermore, AutoGluon achieved an ROC-AUC of 0.8722 and an accuracy of 0.8918, with a high recall (0.9836), indicating a strong sensitivity in detecting the positive class. However, the balanced accuracy (0.6154) suggests an imbalance in performance between the classes, possibly reflecting a greater tendency to correctly predict the majority class. The best model selected by AutoGluon was *RandomForest_BAG_L1*, an independent bagged random forest running at the first stacking level. Unlike more complex ensemble tiers that AutoGluon tested, this first-level model performed best directly on the holdout evaluation structure.

Still regarding AutoGluon, the leaderboard results (Table 3) also show validation consistency among a variety of evaluated base and multi-level ensemble models. The best models presented validation ROC-AUCs between 0.8619 and 0.8803, with *RandomForest_BAG_L1* delivering the most robust final test score.

Model	**ROC-AUC (test)**	**ROC-AUC (val)**	**Stack**	**Fit time (s)**	**Fit order**
WeightedEnsemble_L3	–	**0.8803**	3	1133.94	14
WeightedEnsemble_L2	–	0.8788	2	685.28	7
CatBoost_BAG_L2	–	0.8783	2	788.76	10
LightGBM_BAG_L2	–	0.8782	2	712.76	8
RandomForest_BAG_L1	**0.8722**	0.8690	1	3.09	2
ExtraTrees_BAG_L1	0.8657	0.8619	1	3.36	4

Table 3: Simplified AutoGluon leaderboard (high_quality preset) for vital status classification. "Stack" indicates the stacking level within the ensemble.

Furthermore, the ensemble strategies, separate from AutoGluon, were evaluated using both hard voting and soft voting, as shown in the results in tables 4 and 5. The hard voting ensemble presented an F1-score of 0.8928 and an MCC of 0.4330, showing intermediate performance relative to the best individual models. The soft voting ensemble, on the other hand, achieved an ROC-AUC of 0.8758 and an F1-score of 0.8993, demonstrating an improvement over the individual models regarding ROC-AUC and maintaining a strong balance between metrics.

Model	ROC-AUC	PR-AUC	KS	F1	MCC	Brier	LogLoss
Soft Voting (RF + XGB + MLP)	**0.8758**	**0.9794**	0.5803	0.8993	**0.4441**	0.1116	0.3474
Random Forest	0.8747	0.9792	0.5796	**0.9359**	0.4161	**0.0848**	**0.2804**
XGBoost	0.8741	0.9788	**0.5840**	0.8756	0.4318	0.1359	0.4154
MLP	0.8570	0.9751	0.5453	0.8579	0.3950	0.1472	0.4408

Table 4: Comparison between the soft voting ensemble and the individual models

Model	F1	MCC	ROC-AUC
Hard Voting (RF + XGB + MLP)	0.8928	**0.4330**	–
Random Forest	**0.9359**	0.4161	**0.8747**
XGBoost	0.8756	0.4318	0.8741
MLP	0.8579	0.3950	0.8570

Table 5: Comparison between the hard voting ensemble and the individual models

Finally, Table 6 shows a comparison of the best models for each evaluation metric. It highlights that there is no single superior model across all metrics, reinforcing that the choice of model depends on the metric considered most relevant for the application.

Metric	1st	2nd	3rd	4th
ROC-AUC	Soft Voting	Random Forest	XGBoost	AutoGluon
PR-AUC	Soft Voting	Random Forest	XGBoost	AutoGluon
F1-score	AutoGluon	Random Forest	Soft Voting	Hard Voting
MCC	Soft Voting	Hard Voting	XGBoost	Random Forest
Accuracy	AutoGluon	Random Forest	Soft Voting	Hard Voting
Recall	AutoGluon	Random Forest	Soft Voting	Hard Voting
Brier Score	AutoGluon	Random Forest	Soft Voting	XGBoost

Table 6: Ranking of models according to different evaluation metrics

As observed, the obtained results indicate that all evaluated models showed high performance in classifying the patients' vital status. Although there are some differences among the evaluation metrics, these variations were relatively small, suggesting that random forest, XGBoost, MLP, AutoGluon, and the ensemble approaches constitute viable alternatives for this problem. In this context, the choice of the most appropriate model depends on the prioritized performance criterion, code simplicity, and desired training time.

3.2 Survival analysis models

For the evaluation of time-to-event outcomes and dynamic survival probabilities, two advanced survival regression models were implemented and compared: a Cox-based neural network (DeepSurv) and a gradient-boosted tree model (XGBoost Survival).

Overall, both models showed excellent discriminative performance on the test set, with consistent risk-ranking metrics, as presented in Table 7.

Metric	DeepSurv	XGBoost Survival
C-index (Train)	**0.8598**	0.8318
C-index (Test)	**0.8499**	0.8300
Uno C-index (Test IPCW)	**0.5717**	0.5659
Mean Dynamic AUC	**0.8744**	0.6952
Royston's D	1.2238	**1.3087**
Royston's R^2	0.4766	**0.5101**
Integrated Brier Score (IBS)	0.0778	-
Mean Brier Score	0.0649	-
KS Statistic	**0.5745**	0.5317

Table 7: Comparison of global performance metrics for DeepSurv and XGBoost Survival models.

"–" indicates that the metric was not reported. Calibration metrics based on absolute survival probabilities (e.g., integrated brier score and calibration curves) were not reported for XGBoost Survival because the Cox objective provides relative risk scores rather than calibrated survival probabilities. Therefore, the comparison between models focused on discrimination metrics, which are directly supported by both approaches.

When evaluating baseline generalization and discriminative ability, both models demonstrated excellent stability. DeepSurv achieved superior risk discrimination, with a test C-index of 0.8499 compared with 0.8300 for XGBoost Survival. The overfitting gap remained small for both approaches, reaching 0.0099 for the neural network and only 0.0018 for XGBoost, indicating that neither model exhibited substantial performance degradation between the training and test sets.

For Uno's C-index, which accounts for censoring through inverse probability of censoring weighting (IPCW), both models produced similar results, with DeepSurv achieving a slightly higher value (0.5717 versus 0.5659). A more pronounced difference was observed in the Mean Dynamic AUC, where DeepSurv substantially outperformed XGBoost Survival (0.8744 versus 0.6952), indicating a markedly better ability to discriminate patients according to their time-dependent survival risk. Conversely, Royston's D statistic and its corresponding explained variation (R^2) were slightly higher for XGBoost Survival ($D = 1.3087$, $R^2 = 0.5101$) than for DeepSurv ($D = 1.2238$, $R^2 = 0.4766$). Nonetheless, both models achieved values above 1.0 for Royston's D, indicating satisfactory discrimination.

The discrepancy between a moderate global uno's C-index and a high mean dynamic AUC is a known phenomenon in survival analysis under heavy right-censoring. While uno's C-index evaluates the global ranking of all comparable pairs across the entire follow-up period, its IPCW can become highly unstable due to inflated weights in the extreme tail of the survival distribution. Consequently, minor ordering errors among late-stage events disproportionately penalize the global concordance metric. Conversely, the Mean Dynamic AUC assesses discrimination at specific time horizons, effectively simplifying the evaluation into localized binary classification tasks that are less susceptible to tail-weight instability, thereby reflecting the model's strong performance at clinically relevant time intervals.

Regarding model calibration, the analysis focused on the DeepSurv model, as its neural network formulation natively facilitates the estimation of individual survival probabilities

through the Breslow baseline hazard estimator. DeepSurv indicated robust overall probabilistic calibration and yielded reliable survival probability estimates over time by achieving an Integrated Brier Score (IBS) of 0.0778 and a mean Brier Score of 0.0649.

A critical form of clinical validation consists of stratifying the population according to the log-risk predicted by the models. Table 8 details the division of the test cohort into tertiles (low, intermediate, and high risk), demonstrating the remaining survival probability at the maximum follow-up time evaluated.

Risk Group	**DeepSurv**		**XGBoost Survival**	
	Events	$S(6{,}000$ **d**$)$	**Events**	$S(6{,}000$ **d**$)$
Low Risk ($N = 4897$)	30	0.9925	72	0.9805
Intermediate Risk ($N = 4897$)	283	0.9235	302	0.9104
High Risk ($N = 4898$)	1519	0.6312	1458	0.6446
Log-rank test ($p < 0.001$): DeepSurv $\chi^2 = 1640.78$ | *XGBoost* $\chi^2 = 1588.56$				

Table 8: Kaplan–Meier survival stratification truncated at 6,000 days by predicted-risk tertiles for DeepSurv and XGBoost Survival.

Both models successfully achieved a highly significant separation between the risk-stratified survival curves (Log-rank $p < 0.001$). Patients in the low-risk tertile maintained high survival probabilities in both models, with DeepSurv exhibiting a slightly higher survival than XGBoost Survival at the 6,000-day milestone (0.9925 versus 0.9805, with only 30 versus 72 events, respectively). To ensure statistical reliability and avoid Kaplan-Meier tail artifacts—where cumulative survival estimates become highly unstable due to a severely depleted risk set—the follow-up period for this analysis was truncated at 6,000 days. At this clinical horizon, the survival probability of the high-risk tertile was 0.6312 under DeepSurv and 0.6446 under XGBoost Survival. Crucially, DeepSurv demonstrated superior discriminative power in isolating high-risk patients by concentrating a larger number of fatal events within the high-risk group (1,519 events) compared to XGBoost (1,458 events). This enhanced separation and overall superior clustering of events by DeepSurv are statistically supported by its higher Log-rank statistic ($\chi^2 = 1640.78$ versus $\chi^2 = 1588.56$ for XGBoost). The Kolmogorov–Smirnov (KS) analysis further aligns with these findings, with the maximum separation ($KS = 0.5745$) occurring at the third decile of the DeepSurv model, confirming that the most critical patients are effectively prioritized at the top of the survival risk ranking.

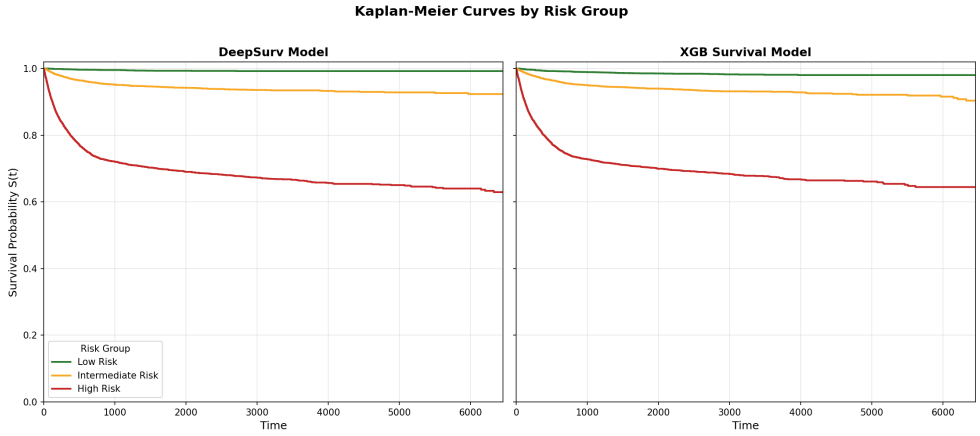

Figure 7: Kaplan-Meier Survival Curves

Figure 7 presents a comparison of the Kaplan-Meier survival curves truncated at 6,000 days of follow-up, stratified by the predicted risk tertiles (low, intermediate, and high) for both the DeepSurv and XGBoost Survival models. Both approaches clearly discriminated the clinical trajectories of the patients across all risk tiers. The decision to truncate the curves at 6,000 days was implemented as a robust methodological safeguard; beyond this timeframe, the sample size of patients remaining under observation decreases drastically, which would introduce high statistical variance and produce misleading tail artifacts in the survival curves. Within this reliable observation window, both models show a highly consistent, parallel decline in their respective high-risk cohorts, which stabilizes after an initial rapid drop, demonstrating excellent agreement between the neural network and the gradient-boosting architectures.

To further investigate the interpretability of the XGBoost Survival model, Figure 8 presents the SHAP beeswarm plot. Unlike global feature importance based solely on information gain, this visualization makes it possible to understand not only which features are most relevant but also the direction and magnitude of their impact on the predicted log-risk for each individual patient. The color gradient (from blue to red) represents the original feature value (from low to high). The horizontal axis (SHAP value) indicates the contribution of each observation to the prediction: positive values (to the right) increase the log-risk of mortality (worse prognosis), whereas negative values (to the left) decrease the risk.

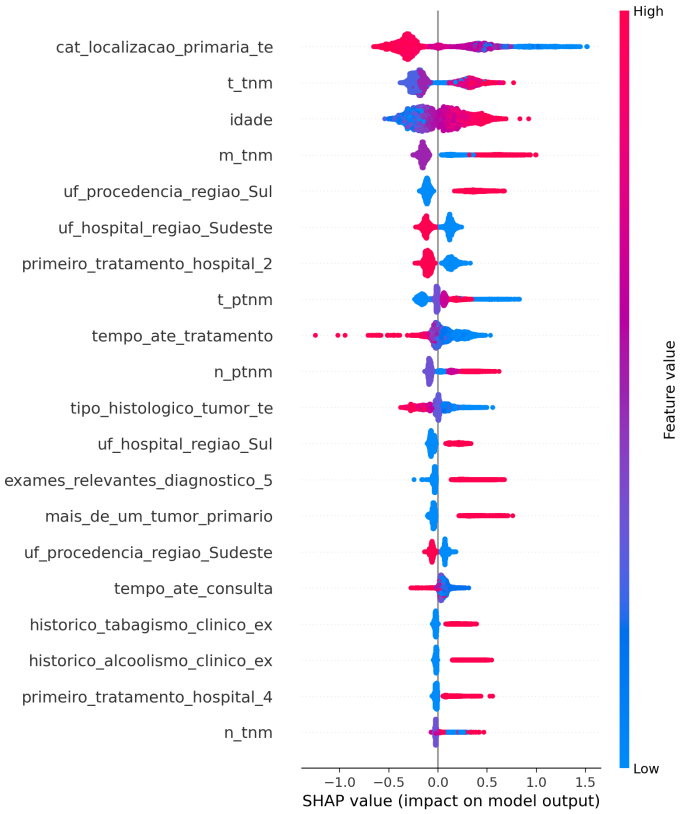

Figure 8: Distribution of SHAP values (beeswarm plot) for the 20 most influential predictor variables in the XGBoost Survival model.

Advanced age (*idade*) and more advanced stages of tumor involvement and metastasis (particularly *t_tnm* and *m_tnm*) exhibit a strong and direct association with increased risk. Red-colored points, representing high values for these variables, shift the SHAP values substantially to the right, indicating a worse temporal prognosis and, consequently, lower expected survival. The primary tumor location category (*cat_localizacao_primaria_te*) emerged as the feature with the greatest overall impact, displaying a wide bilateral dispersion that indicates different anatomical sites substantially alter the patient's baseline risk. Demographic variables, such as origin from the Southern region (*uf_procedencia_regiao_Sul*), also ranked among the most influential predictors, reinforcing the contribution of geographic or regional systemic factors to clinical outcomes.

Additionally, the model captured a counterintuitive association that may reflect confounding by indication regarding the time to treatment (`tempo_ate_tratamento`), where shorter waiting times (blue points) were associated with higher risk. This pattern likely reflects a confounding-by-indication bias, where high-urgency, critical patients are prioritized for rapid clinical intervention but naturally present worse survival prognoses.

4 Conclusion

Cancer represents one of the most pressing public health challenges in Brazil, requiring data-driven strategies to improve epidemiological surveillance and optimize resource allocation within the Unified Health System (SUS). The central focus of this study was to extract value from and address the data quality challenges present in the longitudinal oncology records of the National Cancer Institute (INCA), a database that initially contained approximately 5.4 million records. The objective was to develop predictive models capable of anticipating patients' vital status and survival time, thereby creating a clinical decision support tool able to handle temporal inconsistencies, the high proportion of missing values in key clinical staging variables, and the class imbalance inherent to medical data.

To address these challenges, the methodology was based on the development of a data processing pipeline using PySpark within the Databricks environment. This stage involved rigorous validation of treatment chronology, handling of missing values, decomposition of clinical staging abbreviations (TNM/pTNM), and the engineering of five new features focused on disease duration. After applying domain-specific consistency filters, the dataset was refined to 58,767 training samples and 14,692 test samples. The modeling phase was then divided into two approaches: binary classification algorithms (random forest, XGBoost, MLP neural network, AutoGluon, and ensemble models) to predict static vital status using class weighting to address class imbalance; and survival analysis models (XGBoost Survival and DeepSurv) designed to predict time to death while naturally handling right-censored data.

The results demonstrated the feasibility and effectiveness of applying machine learning to Brazilian hospital cancer registry data, with all models exhibiting high discriminative performance. In the classification task, the algorithms achieved ROC-AUC values ranging from 0.86 to 0.88, with AutoGluon obtaining the highest F1-score (0.9406) and achieving the best overall balance (PR-AUC of 0.9782, KS of 0.5848, MCC of 0.4441, Brier of 0.0798, and Log-Loss of 0.2639). In the survival analysis, both algorithms produced a clear separation of the population survival curves into distinct risk groups ($p < 0.001$); however, the DeepSurv neural network demonstrated a greater ability to capture the temporal dependence of risk (Mean Dynamic AUC of 0.8744) compared with XGBoost. Finally, model interpretability revealed that features such as the primary tumor location category, tumor metastasis (TNM) and patient age in Random Forest or histological tumor type in XGBoost had the greatest predictive influence on the evaluated outcomes.

Future work may address limitations inherent to large-scale clinical registry data. The outcome records in the present dataset are constrained by incomplete longitudinal follow-up, as reliable information is largely limited to recorded deaths, without a systematic last-contact date for censored patients; access to more comprehensive follow-up data or linkage with external mortality registries would reduce this source of uncertainty and support a more accurate characterization of patient survival. Relatedly, the binary classification framework could be reformulated as survival prediction within predefined clinical horizons (e.g., 3- or 5-year survival), facilitating direct comparison with existing prognostic studies. The absence of reliable patient-level identifiers also precluded subject-wise data partitioning, meaning that multiple records from the same individual could not be guaranteed to remain within a

single subset; datasets with such identifiers would allow this issue to be addressed directly in future analyses.

Beyond these dataset-specific constraints, validating the proposed models on external cohorts from different healthcare institutions and geographical regions remains an essential next step to assess their generalizability and robustness. Additional efforts may also explore the incorporation of longitudinal clinical information, molecular and genomic biomarkers, and imaging-derived features to further improve predictive performance. Finally, prospective clinical validation and integration into decision support systems could facilitate the practical adoption of these models in routine oncology care, enabling more personalized risk stratification and supporting clinical decision-making in real-world settings.

References

[1] AutoGluon Contributors. *AutoGluon Documentation*. https://auto.gluon.ai. Accessed: May 7, 2026. 2024.

[2] Leo Breiman. "Random forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[3] Glenn W. Brier. "Verification of forecasts expressed in terms of probability". In: *Monthly Weather Review* 78.1 (1950), pp. 1–3. DOI: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

[4] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[5] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC Genomics* 21.1 (2020), p. 6. DOI: 10.1186/s12864-019-6413-7.

[6] T. G. Clark et al. "Survival analysis part I: basic concepts and first analyses". In: *British Journal of Cancer* 89.2 (2003), pp. 232–238. DOI: 10.1038/sj.bjc.6601118.

[7] David R. Cox. "Regression models and life-tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972), pp. 187–202. DOI: 10.1111/j.2517-6161.1972.tb00899.x.

[8] Cynthia S. Crowson, Elizabeth J. Atkinson, and Terry M. Therneau. "Assessing calibration of prognostic risk scores". In: *Statistical Methods in Medical Research* 25.4 (2016), pp. 1692–1706. DOI: 10.1177/0962280213497434.

[9] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems* (2000), pp. 1–15.

[10] Nick Erickson et al. *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data*. 2020. arXiv: 2003.06505 [stat.ML]. URL: https://arxiv.org/abs/2003.06505.

[11] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[12] Thomas R. Fleming et al. "Modified Kolmogorov-Smirnov test procedures with application to arbitrarily right-censored data". In: *Biometrics* 36.4 (1980), pp. 607–625. DOI: 10.2307/2556114.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge: MIT Press, 2016. URL: https://www.deeplearningbook.org/.

[14] Erika Graf et al. "Assessment and comparison of prognostic classification schemes for survival data". In: *Statistics in Medicine* 18.17-18 (1999), pp. 2529–2545. DOI: 10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5.

[15] Sunil Gupta et al. "Machine-learning prediction of cancer survival: a retrospective study using electronic administrative records and a cancer registry". In: *BMJ Open* 4.3 (2014). ISSN: 2044-6055. DOI: 10.1136/bmjopen-2013-004007.

[16] Frank E. Harrell et al. "Evaluating the yield of medical tests". In: *JAMA* 247.18 (1982), pp. 2543–2546. DOI: 10.1001/jama.1982.03320430047030.

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer Science & Business Media, 2009. DOI: 10.1007/978-0-387-84858-7.

[18] Patrick J. Heagerty and Yuanlong Zheng. "Survival model predictive accuracy and ROC curves". In: *Biometrics* 61.1 (2005), pp. 92–105. DOI: 10.1111/j.0006-341X.2005.030814.x.

[19] Instituto Nacional de Câncer. *Estimativa 2023: incidência de câncer no Brasil*. INCA, 2022. URL: https://www.inca.gov.br/publicacoes/livros/estimativa-2023-incidencia-de-cancer-no-brasil.

[20] Instituto Nacional de Câncer. *Registros Hospitalares de Câncer: Planejamento e Gestão*. 2nd. INCA, 2010. URL: https://www.inca.gov.br/publicacoes/manuais/registros-hospitalares-de-cancer.

[21] Peng Jiang et al. "Big data in basic and translational cancer research". In: *Nature Reviews Cancer* 22 (2022), pp. 625–639. DOI: 10.1038/s41568-022-00502-0.

[22] Tejasvi Sanjay Kamble et al. "Predicting cancer survival at different stages: Insights from fair and explainable machine learning approaches". In: *International Journal of Medical Informatics* 197 (2025), p. 105822. ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2025.105822.

[23] Edward L. Kaplan and Paul Meier. "Nonparametric estimation from incomplete observations". In: *Journal of the American Statistical Association* 53.282 (1958), pp. 457–481. DOI: 10.1080/01621459.1958.10501452.

[24] Jared L. Katzman et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network". In: *BMC Medical Research Methodology* 18.1 (2018), p. 24. DOI: 10.1186/s12874-018-0482-1.

[25] David G. Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text*. 3rd. New York: Springer, 2012. DOI: 10.1007/978-1-4419-6646-9.

[26] Ron Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *14th International Joint Conference on Artificial Intelligence (IJCAI)*. 1995, pp. 1137–1143.

[27] Konstantina Kourou et al. "Machine learning applications in cancer prognosis and prediction". In: *Computational and Structural Biotechnology Journal* 13 (2015), pp. 8–17. DOI: 10.1016/j.csbj.2014.11.005.

[28] Nathan Mantel. "Evaluation of survival data and two new rank order statistics for their comparison". In: *Cancer Chemotherapy Reports* 50.3 (1966), pp. 163–170.

[29] Brian W. Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451. DOI: 10.1016/0005-2795(75)90109-9.

[30] M. Piñeros et al. "Cancer registration for cancer control in Latin America: a status and progress report". In: *Revista Panamericana de Salud Pública* 41 (2017). DOI: 10.26633/RPSP.2017.2.

[31] David M. W. Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63.

[32] Anjali Raghav, Sharad Vaish, and Monika Gupta. "Survival Prediction of Cancer Patient Using Machine Learning". In: *Concepts and Real-Time Applications of Deep Learning*. Ed. by Smriti Srivastava et al. Springer International Publishing, 2021. ISBN: 978-3-030-76167-7. DOI: 10.1007/978-3-030-76167-7_6.

[33] Patrick Royston and Willi Sauerbrei. "A new measure of prognostic discriminatory power in survival analysis". In: *Statistics in Medicine* 23.5 (2004), pp. 723–748. DOI: 10.1002/sim.1621.

[34] Shawn M. Sweeney et al. "Challenges to Using Big Data in Cancer". In: *Cancer Research* 83 (2023), pp. 1175–1182. DOI: 10.1158/0008-5472.can-22-1274.

[35] Hajime Uno et al. "On the C-statistics for evaluating overall adequacy of risk prediction models with censored survival data". In: *Statistics in Medicine* 30.10 (2011), pp. 1105–1117. DOI: 10.1002/sim.4154.

[36] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.