Classification of Breast Cancer Patients into Survivors or Not

Master in Data Science, FIB
Machine Learning

June 12th, 2023

Marçal García Boris - Odysseas Kyparissis

Table of Contents

Description and Goals

We all know that Machine Learning (ML) and Artificial Intelligence (AI) technologies have revolutionized today's world, by affecting most of the important scientific fields that play a crucial role in society's evolution. One of those fields is Medicine and Healthcare. By analyzing big amounts of data, ML and AI models can provide great support to healthcare professionals in creating accurate diagnoses and providing personalized treatment plans. In addition, it is very inspiring that with the usage of such technologies scientists can predict the occurrence of critical diseases (e.g. Breast Cancer) and thus prevent as soon as possible the outrage of such diseases. For that reason, we decided to choose the **Breast Cancer Dataset** which involves female patients with infiltrating ductal and lobular carcinoma breast cancer diagnosed in 2006-2010.

To begin with, the dataset consists of **16 variables** (5 numerical and 11 categorical) and **a total of 4024 observations**. The numerical variables of the dataset included the patient's *Age*, *Tumor Size* measured in millimeters, the number of examined lymph nodes (*Regional Node Examined***)**, the count of lymph nodes that tested positive for cancer cells (*Regional Node Positive***)**, and the survival duration in months since diagnosis (*Survival Months*).

On the other hand, the categorical variables included the patient's *Race*, categorized as *White*, *Black*, or *Other* (American Indian/ AK Native, Asian/ Pacific Islander), as well as their *Marital Status*, which could be *Divorced*, *Married*, *Separated*, *Single*, or *Widowed*. Other categorical features captured aspects such as the size of the primary tumor (*T Stage*), the extent of lymph node involvement (*N Stage*), the spread of cancer cells and tumor size (*6th Stage*), tumor differentiation (*differentiate*), cancer grade, indicating aggressiveness (Grade), early stage and spread (*A Stage*), estrogen receptor status (*Estrogen Status*), progesterone receptor status (*Progesterone Status*), and the target variable indicating the patient's *Status* as *Dead*' or *Alive*. These categorical variables provide valuable insights into the various characteristics and attributes associated with the patients and their cancer diagnosis.

Finally, the structure of the report consists of the following main sections: Exploratory Data Analysis (EDA), Modeling and Conclusion. In more detail, the EDA section covers topics such as: Univariate Exploratory Analysis, Univariate Outliers Analysis, Bivariate Exploratory Analysis, Multivariate Outlier Analysis, and Feature Selection/Extraction, while the Modeling section includes Dataset Splits, Preprocessing, Learning Algorithms, Cross Validation, Model Comparison & Hyper Parameter Tuning, and the Final Model sections.

Exploratory Data Analysis (EDA)

In this section of the report, the implementation of the exploratory analysis, both univariate and bivariate, as well as the outlier detection procedure is presented. Finally, the section concludes with the feature selection and extraction process.

Univariate Exploratory Analysis

In order to get a first overview of the dataset, it is necessary to analyze each individual variable available. Firstly, it is important to mention that there are no missing values included in the dataset. However, we need to take a look at the distribution of the numerical variables, as well as the categories of the categorical variables in order to be sure that the dataset does not contain erroneous data. After careful consideration, inside the *notebooks* provided as the solution of this project, it is concluded that erroneous data do not exist.

Furthermore, we split our data into *numerical* and *categorical* variables to proceed with the EDA. Once this is done, it is possible to create *histograms* in the case of *numerical* variables and *barplots* for the *categorical* ones. Figure 1. depicts the distributions of the variables.

Upon closer examination of the variables, it becomes evident that there is a considerable imbalance among the categorical variables. For instance, the majority of women in the dataset are categorized as *White*, while the other racial groups have significantly fewer observations. Similarly, a large proportion of the women are *Married*, with only a minimal number being *Separated*. Regarding tumor characteristics, *T1* and *T2* stages dominate the available data, whereas cases of *T3* and *T4* are relatively less represented. In addition, within the *N Stage* categorical variable, *N1* observations are predominant, suggesting a potential imbalance in lymph node involvement. The *6th Stage* variable exhibits a lack of observations specifically in the *IIIB* category. Notably, the *differentiate* variable is heavily skewed, as the undifferentiated category comprises only a small fraction of the dataset. A similar pattern emerges for the *Grade* variable, particularly the *anaplastic; GradeIV* category. Moreover, the *A Stage* variable indicates a limited number of cases classified as *Distant*. The *Estrogen and Progesterone Status* variables also demonstrate imbalances, particularly in terms of the negative values. Finally, the distribution of the target variable, which indicates patient *Status*, is unbalanced, with a higher number of observations for patients classified as *Alive* compared to those classified as *Dead*.

Turning to the numerical variables, the *Age* distribution appears to be relatively well-distributed, with observations spanning various ages between 30 and 70 years old. In contrast, the distribution of *Tumor Sizes* skews towards the left, indicating a concentration of smaller values, although there are a few instances of significantly larger tumor sizes, indicating the presence of potential outliers that should be appropriately addressed. The distribution of the *Regional Node Examined* variable is reasonably balanced, with only a few extreme cases at lower values and some outliers in the higher range. Similarly, the *Regional Node Positive* variable exhibits a left-skewed distribution, suggesting that a majority of observations have low values, while a minority exhibit high values. Lastly, the *Survival Months* variable appears to be slightly right-skewed, implying that specific treatment approaches may be necessary for observations with smaller numbers of survival months (ranging from 0 to 40) to ensure accurate analysis and interpretation. The insights generated by the univariate analysis need to be addressed in order to avoid fitting a learning algorithm which will generate a model described by high bias.

Before moving on to the Univariate Outliers Analysis, the Shapiro-normality[1] test is performed on the numerical values. In that way it can be statistically proven which numerical variables follow a normal distribution.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html

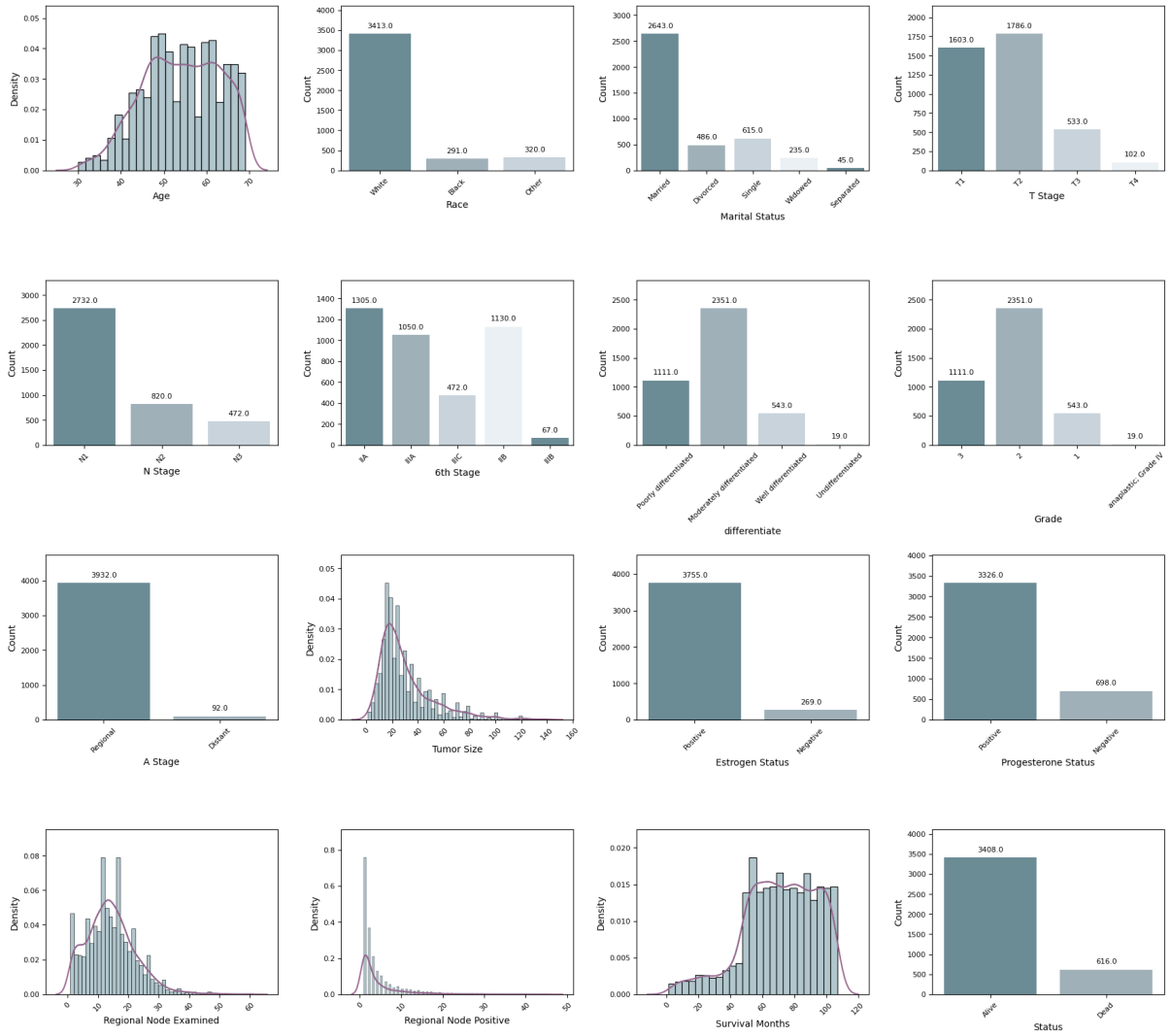


Figure 1. Univariate Analysis

Later in the Bivariate Exploratory Analysis, Chi-Squared tests are performed on the categorical variables as well, in order to check correlation between them. The results reveal that **none of the numerical variables** (*Age*, *Tumor Size*, *Regional Node Examined*, *Regional Node Positive*, and *Survival Months*) **exhibit a normal distribution**, as indicated by the extremely small p-values obtained (results in the *notebooks)*. This suggests that these variables deviate significantly from a normal distribution. To mitigate potential issues during the modeling phase, it is considered to apply suitable transformations or utilizing non-parametric models that do not rely on normality assumptions. Although, by taking into account the distribution and the statistics of *Regional Node Examined, Regional Node Positive* and *Survival Months*, the values seem reasonable. However *Tumor Size* variable needs specific treatment. After several tries it was concluded that *BoxCox*[2] is the appropriate transformation for the specific variable. However, the transformation is taking place in the Preprocessing section, since *BoxCox* calculates the *lambda* value of the transformation by taking into account the total distribution of the variable. For such a reason, it is crucial to first split the data into *training* and *testing* sets and then proceed with applying this transformation.

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.boxcox.html

Univariate Outliers Analysis

During the development and training of learning algorithms the inclusion of outliers can result in high bias results, thus it is very important to treat them carefully, first by identifying them and finally by imputing or removing their values, or just by separating them from the remaining observations, in order to treat them separately. The analysis is completed for all the numerical variables of the dataset.

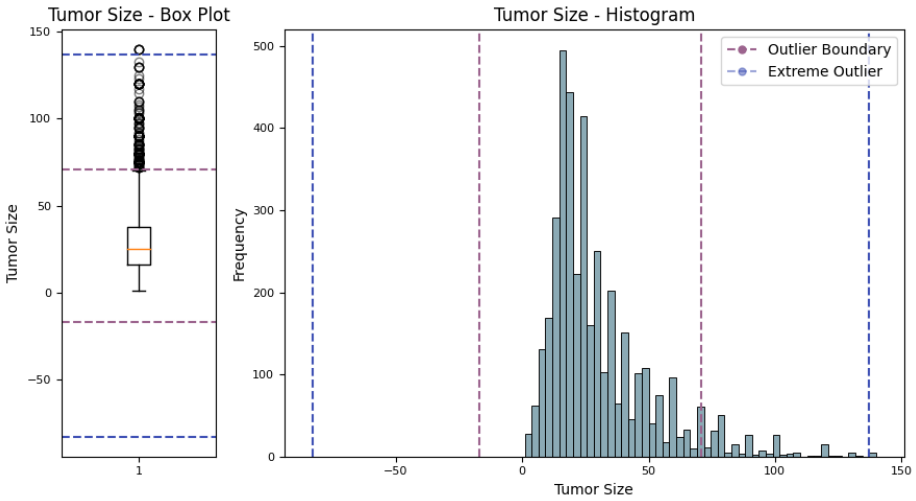


Figure 2: Outlier Analysis in Tumor Size

Figure 2. presents an example for the *Tumor Size* variable. As you can see, the purple and blue dotted lines indicate the *Q1* and *Q3* respectively, following the interquartile range (IQR)[3] approach. Thus, all the values between the first and the second boundary will be considered as outliers, while those located at the extremes will be considered extreme outliers. For the remaining *numerical* variables (except *Age*), the results are depicted in Appendix.

As we can see, all the variables present some kind of outliers. For the variable *Tumor Size* we can see that there are *222* outliers in total (starting from values of 71), from which 5 are considered extreme outliers (having size equal to 140). Specifically, by carefully inspecting the 5 extreme cases of the variable, the combination of the information included for those observations, seemed quite abnormal, and for that reason, it was decided to separate them from the dataset, and analyze them separately. The analysis is continued by including *Regional Node Examined*. For this variable it can be seen that there are not any extreme outliers, while only 72 values are considered as outliers. If we move on to *Regional Node Positive*, we can see that there are 54 extreme outliers (with values more than 23), while 344 values are considered as outliers. Lastly, Survival Months are analyzed. For those 3 variables, the total set of observations was retained. More detailed information can be found in the solution's *notebooks.*

[3] https://en.wikipedia.org/wiki/Interquartile_range

Bivariate Exploratory Analysis

In this section, a deeper look is taken into the relationship between the pairs of available variables with respect to the target variable *Status*. Firstly, the relations between the numerical variables with respect to the target variable is depicted in Figure 3..

At first sight, by taking a look at the scatter plot of the numerical variables, one can understand that the observations of ***Alive*** and ***Dead*** classes are mixed when a combination of two numerical variables occurs. This means that there is no clear separation between a pair of numerical variables for distinguishing ***Alive*** from ***Dead*** cases. One interesting insight from the scatterplot occurs for the variable *Survival Months*, which seems to generate a good separation boundary between the two classes, with all the combinations of the remaining numerical variables. This insight leads to the conclusion that the variable *Survival Months* can offer quite strong predictability power to the model.

Moreover, inside the *notebooks,* a look at the correlation[4] heatmap is taken at this point (see Appendix). The result of the heatmap introduces the insight that the correlation between *Regional Node Examined* and *Regional Node Positive* indicates a moderate positive correlation of **0.412**, thus, as the number of regional nodes examined increases, the number of positive nodes also tends to increase. Later on, it is shown that the generation of a new feature which includes the ratio of those 2 variables, provides great power to the analysis and the model.

To continue with. a deeper look into the distribution of the categorical variables with respect to the target variable is taken, and it is presented in Figure 4.. By considering the bar plots of Figure 4., there are not clear conclusions to be made, due to the unbalanced nature of the dataset as mentioned earlier.

However, some interesting insights generated by this figure is that for the variables *T Stage*, *N Stage* and *6th Stage*, it can be observed that for the categories ***T4***, ***N3*** and ***IIC***, respectively, the difference between the ***Alive*** and ***Dead*** cases, tend to get smaller, which might mean that these specific categories will help the model recognise patterns for ***Dead*** cases. Finally, it can be observed that in some specific categories, the probability of survival increases, for example, when the patient has a ***T1*** value for the *T Stage* variable.

At this point it is necessary to statistically test if correlation exists between the categorical variables, and this is accomplished by using the Chi Squared test. By taking a look at the final results of the tests (found inside *notebooks* and ***chi-2.csv*** file delivered), the following conclusions can be derived.

Several significant associations exist among the categorical variables in the dataset. For example, *Race* and *Marital Status* are significantly associated (Chi-square = 137.96, $p < 0.001$). *Race* also shows significant associations with *differentiation*, *Grade*, *Estrogen Status*, and *Status*. *Marital Status* is significantly associated with *T Stage*, *N Stage*, *differentiation*, *Grade*, and *Status* as well. *T Stage* and *N Stage* strongly correlate with each other and are significantly associated with multiple variables. *Differentiation*, *Grade*, *A Stage*, *Estrogen Status*, *Progesterone Status*, and *Status* also exhibit significant associations with other variables. Overall, these associations provide valuable insights into the

[4] https://en.wikipedia.org/wiki/Correlation

interconnections between different variables and their impact on the final *Status* of the individuals in the study. However, since great association exists between the categorical variables, an approach of dimensionality reduction, could help compressing this information into as few variables as possible, in order to reduce the complexity and size of the final model and analysis. Discussion about this topic is handled in section Interpretability of the Final Model.

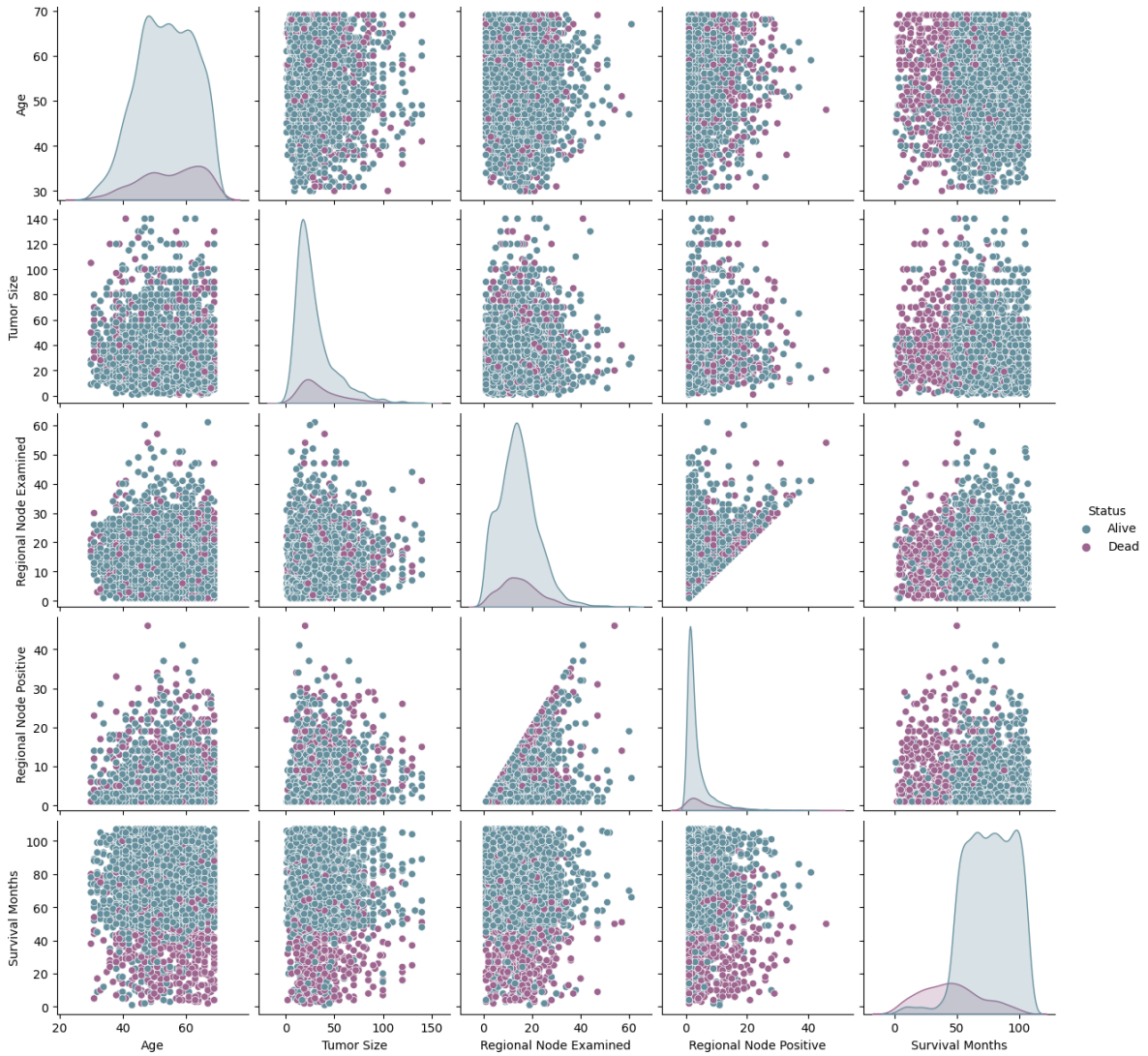
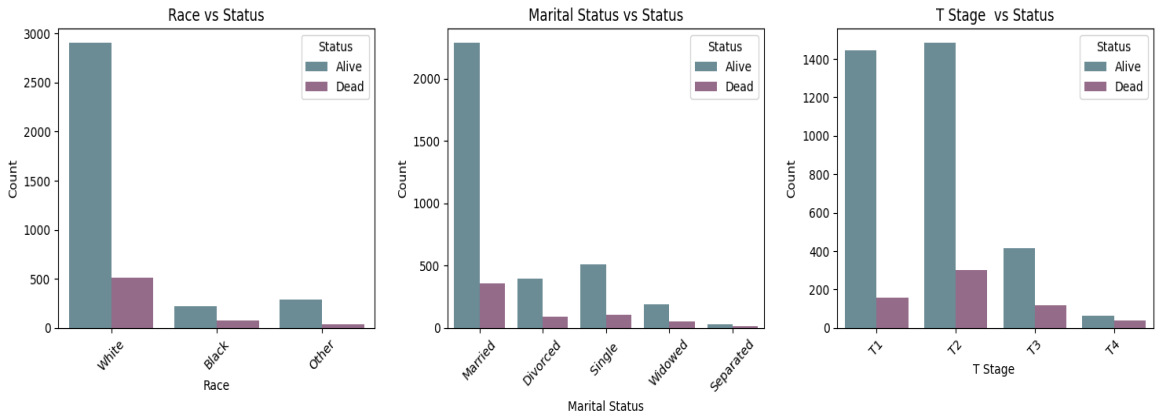


Figure 3: Bivariate Scatterplot Analysis with Respect to *Status*



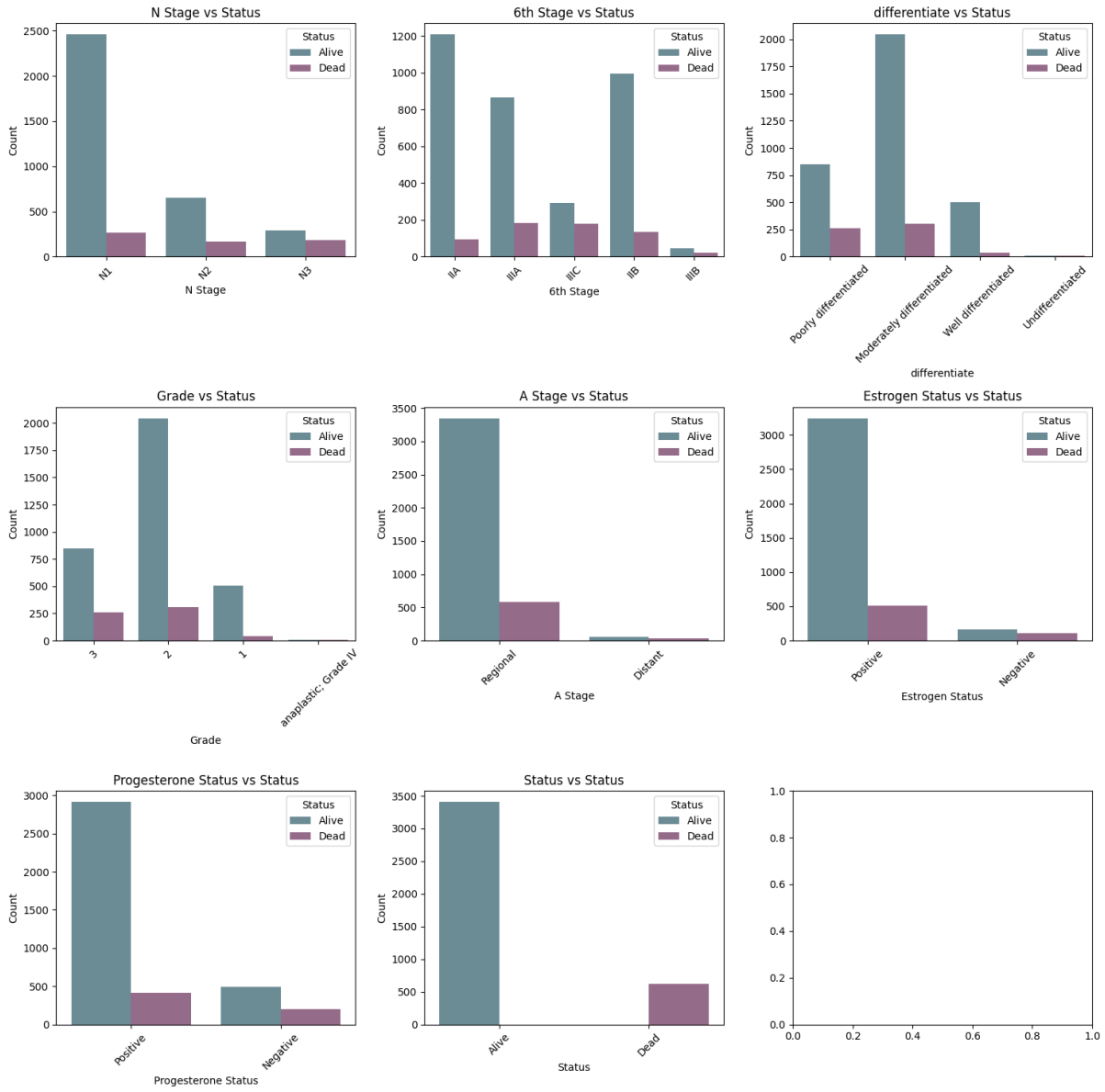


Figure 4: Bivariate Barplot Analysis with Respect to *Status*

Multivariate Outlier Analysis

The next step of the EDA is the multivariate outlier analysis. A multivariate outlier detection was performed using *Mahalanobis Distance*[5] on the numerical variables of the dataset. In order to do so the calculation of the *covariance matrix*[6] and its inverse is necessary in order to calculate the distances between the observations of the dataset. The cutoff value for detecting outliers is set to **1%** based on the *Chi-Square* distribution. Thus, **0.99** represents the desired significance level. Figure 5. depicts the distribution of the data with respect to the Mahalanobis Distance.

[5] https://en.wikipedia.org/wiki/Mahalanobis_distance
[6] https://en.wikipedia.org/wiki/Covariance_matrix

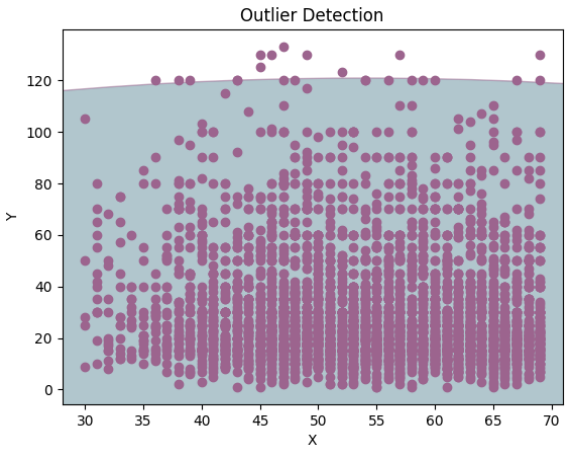


Figure 5: Mahalanobis Distance Distribution, Multivariate Outliers

Based on the analysis mentioned, there are **177** multivariate outliers. After careful consideration and by analyzing the statistics of those observations (details in *notebooks*), we consider retaining them inside the dataset, since the distribution of their values seem logical. Although, we might return to this point in case of failures during the modeling phase.

Feature Selection/Extraction

As previously mentioned, the dataset is quite imbalanced. Consequently, most of the levels of one variable dominate the majority of the observations. We aimed to address this issue by replacing the original variables with ones that have fewer levels derived from them. After considering all the variables, we made the decision to create five new features: three categorical and two continuous.

Regarding *Age*, the dataset was divided into **three groups**: *Young*, *Middle-aged*, and *Elderly*. This categorization effectively reduced the number of distinct values associated with *Age*. Considering the racial distribution, there was a significant disparity among the patient counts for different races. To mitigate this, we introduced a binary variable indicating whether a patient is *Caucasian* or *Non-Caucasian*. This adjustment helped narrow the gap between the original three racial categories: *White*, *Black*, and *Other*. Moreover, In terms of *Tumor Stage*, the data showed a strong imbalance, with *T1* and *T2* stages comprising the majority of cases. To address this, the stages were grouped into two categories: *Early Stage (T1)* and *Advanced Stage (T2, T3, and T4)*. This grouping achieved a more balanced distribution, splitting the data nearly evenly between the two new levels.

Finally, two new continuous variables were also introduced. The first one, *Age at diagnosis*, captures the age of the patient at the time of diagnosis. The second continuous variable, *Positive Lymph Ratio*, was derived from the number of examined lymph nodes and the number of lymph nodes that tested positive. After analyzing the final version of the dataset, it can be seen that *Positive Lymph Ratio can* replace the occurrence of the two numerical variables used to derive this new feature, without including any information loss (see *modeling notebook*).

To conclude on the EDA, before moving on to the Modeling section, the univariate, bivariate and confusion matrix figures were regenerated for the new version of the dataset, and they can be found inside the *notebooks* and the Appendix.

Modeling

In this section the methodology of preprocessing the data, as well as, the designing, training and testing of several learning algorithms is described. Once the comparison between the different models and their hyper parameters is completed, the *best* model is selected as the solution of this project. Finally, a generalization analysis of the final model is included.

Dataset Splits

In order to start the modeling process it is very important to split the dataset into training and testing sets. Two data partitions (**train** and **test**) are sampled and later on we will apply cross-validation over the train partition for conducting model comparison and deciding hyperparameters.

Creating separate training and test sets is essential to ensure an unbiased evaluation of the model's performance. The training set is used to build and optimize several models, compare them and thus select the most appropriate, while the test set acts as an unseen dataset for assessing the generalization ability of the *best* model. By doing so, a realistic estimate can be obtained of how well the model generalizes and performs on unseen data.

In addition, employing cross-validation on the training set offers several advantages: helps to robustly tune the model's hyperparameters by iteratively training and evaluating the model on different subsets of the training data. This process helps to identify the best hyperparameter configuration that maximizes performance and minimizes overfitting. Cross-validation provides a more reliable estimate of the model's performance, reducing the risk of bias introduced by a single train-test split.

Preprocessing

For the preprocessing, we will normalize the numerical variables with the usage of the ***Min-max Scaler***[7], and finally ***One-hot encoding***[8] for the categorical variables. It is very important to distinguish the preprocessing steps between the two sets, in order to avoid introducing bias. The goal is to simulate the testing of the model, with unseen data, as it will happen in a real-case scenario. The implementation of the preprocessing process shows that the scaler is fitted only with the data of the training set. Then this scaler is applied to the testing data as well. As we know, numerical transformation could be done differently, for example, using **standardization** (*Standard Scaler*[9]). Nevertheless, in our case using this kind of normalization leads to a drop in our final results.

During the EDA it was shown that the variable *Tumor Size* could be transformed into a normal distribution using BoxCox transformation, but we needed to be cautious about it and

[7] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

do the transformation separately from the training and test datasets. It is important to mention that, after generating the final model, it is crucial to automate the process of preprocessing of the data, in order to be applied to the new unseen data so that they can be ingested by the model to produce the prediction of the target variable. Figure 6. presents the transformation of *Tumor Size*.

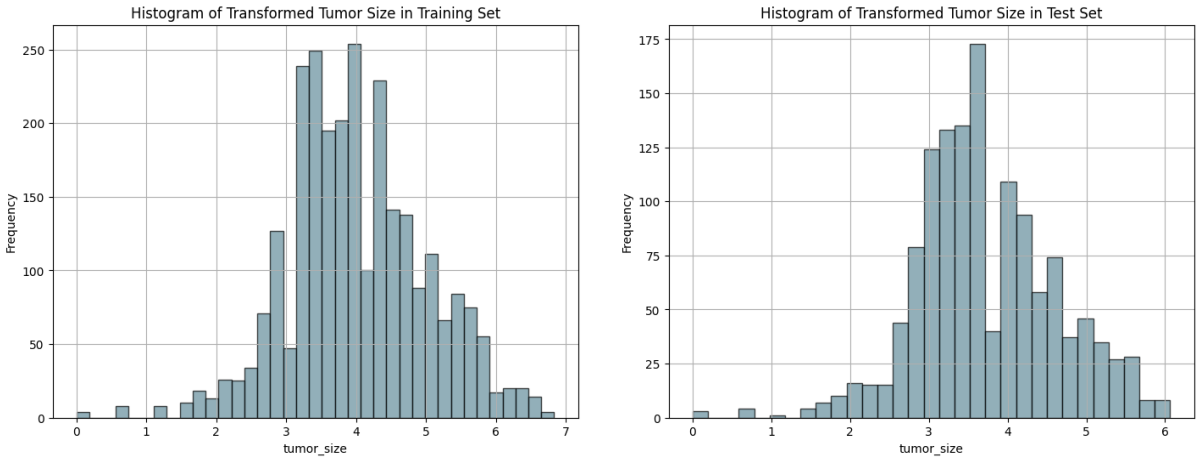


Figure 6: Tumor Size Transformation for Training (left) and Test (right) Sets

Learning Algorithms

In order to decide which is an appropriate learning algorithm, for the specific dataset, that can satisfactorily predict the binary outcome, different classification models were tried. The selected models for this process were the following: **Logistic regression**[10], **Random forest**[11], **Support Vector Machine**[12] (**SVM)**, **Gradient boosting**[13], **k-nearest neighbors**[14] (**KNN)**, **Decision trees**[15] and **Naive Bayes**[16]**.** The selection of the mentioned algorithms seemed reasonable, since they can handle two-class outcomes, accommodate both numerical and categorical features, and capture complex relationships between predictors and the target variable, thus enabling accurate predictions in this medical scenario. Each algorithm has its strengths, ranging from logistic regression's interpretability to Random forest's robustness and SVM's ability to handle high-dimensional data, allowing for comprehensive exploration and selection based on specific requirements.

Model Comparison & Hyper Parameter Tuning

As mentioned in the Dataset Splits section, performing cross-validation[17] is beneficial for model comparison and hyperparameter tuning, since an iterative training and evaluating process is completed for each separate learning algorithm using different training subsets. Before explaining the procedure followed, it is crucial to mention that *Random*

[10] https://en.wikipedia.org/wiki/Logistic_regression
[11] https://en.wikipedia.org/wiki/Random_forest
[12] https://en.wikipedia.org/wiki/Support_vector_machine
[13] https://en.wikipedia.org/wiki/Gradient_boosting
[14] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[15] https://en.wikipedia.org/wiki/Decision_tree
[16] https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[17] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Oversampling[18] was used in order to balance the positive and negative examples of the training set before fitting the algorithms. Without applying this process, the predictability results of the models were very poor, although it's essential to be cautious about the potential drawbacks of oversampling, such as overfitting and increased computation time.

For each learning algorithm, we employ GridSearchCV[19], which combines grid search with cross validation, to identify the optimal hyperparameters. In our case, the grid search with cross-validation mentioned before already takes care of internally splitting the training data into training and validation subsets for each fold of cross-validation, thus, it is not necessary to previously create the validation subsets from the original data. The process then finds the best hyperparameters based on the mean performance metrics (*Accuracy*[20], *Precision*[21], *Recall*[22], *F1-score*[23]) calculated on the validation subsets. In that way, the final evaluation of the selected model on the test set will provide an unbiased estimate of the model's performance and generalization. Figure 7. presents the training scores for each of the models.

From the results of the cross-validation procedure, it is concluded that the most appropriate model for the specific use-case seems to be the ***Random Forest*** algorithm which achieves the highest value of *Recall* and *F1-score* respectively. It is really important, in this case, **to achieve a high *Recall* value**, since the model's goal is to classify unseen (future) patients with Breast Cancer into **Alive** or **Dead** cases, thus it is important to minimize the misclassification of truly *Dead* cases into falsely predicted *Alive* labels. Consequently, we can sacrifice the performance on the rest of the metrics, focusing solely on *Recall*.

Model	Best Hyperparameters	Accuracy	Precision	Recall	F1-score	Validation Errors
Logistic Regression	{'C': 1, 'class_weight': None, 'penalty': 'l1'}	0.797103	0.792764	0.804576	0.79855	0.010134
Random Forest	{'class_weight': 'balanced', 'max_depth': 10, ...	0.915019	0.909497	0.921833	0.915568	0.011617
SVM	{'C': 10, 'class_weight': 'balanced', 'kernel'...	0.855073	0.864324	0.842774	0.853282	0.006828
Gradient Boosting	{'learning_rate': 0.1, 'n_estimators': 300}	0.901186	0.913674	0.886262	0.899721	0.013388
KNN	{'n_neighbors': 3, 'weights': 'distance'}	0.901844	0.998383	0.805011	0.891281	0.005844
Decision Tree	{'class_weight': None, 'max_depth': 10, 'min_s...	0.87725	0.89786	0.852016	0.873843	0.013492
Naive Bayes	{}	0.684454	0.639645	0.845414	0.728181	0.00606

Figure 7: Training Scores

Final Model

Once the **Random Forest** algorithm is selected as the final model, we need to be sure that the model is not over or underfitting the data. From the previous analysis, the best hyper parameters for the model are the following: *class_weight* = 'balanced', *max_depth* = 10, *n_estimators* = 200. To accomplish that, it is necessary to check the training and test errors with respect to several parameters.

[18] https://imbalanced-learn.org/stable/over_sampling.html
[19] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV
[20] https://en.wikipedia.org/wiki/Accuracy_and_precision
[21] https://en.wikipedia.org/wiki/Accuracy_and_precision
[22] https://en.wikipedia.org/wiki/Precision_and_recall
[23] https://en.wikipedia.org/wiki/F-score

Training-Test Error Analysis on the max_depth Parameter

In this subsection, Figure 8. obviously presents that when the parameter *max_depth* of the model exceeds a certain threshold (approx. **11**), the model is overfitting the data, leading to very small training error while the test error remains steady, or slightly increases. For that reason, we will try to train the algorithm with *max_depth* values equal to around **8-10** which seems *just-right*.

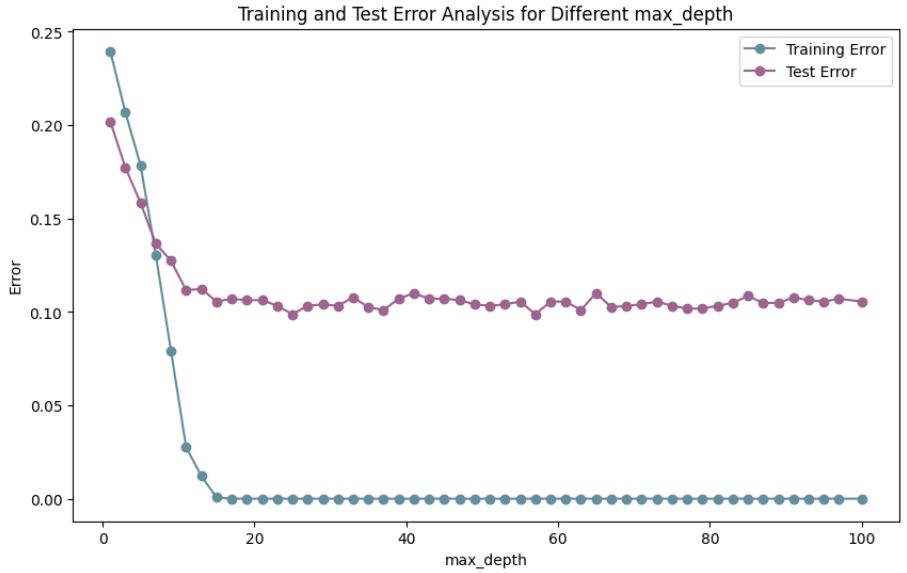


Figure 8: Training and Test Error for Different *max_depth* Values

Training-Test Error Analysis on the n_estimators Parameter

As for the *n_estimators* parameter of the model, the result of Figure 9. is not so obvious. It seems that over the whole range of the parameter, from ***5*** to ***500***, both the training error and the test error remain approximately constant. Thus a clear conclusion cannot be made from this figure.

Figure 9: Training and Test error for different number of estimators

Training-Test Error Analysis on the Training Data Size

By taking a look at Figure 10., one can understand that as the *Training Set Size* which is fed into the model increases, the training misclassification error increases but very slightly, since the worst error achieved is approx. **0.06**. As for the test misclassification error, it is clear that it fluctuates between the values **0.10** and **0.12**. However, the slopes of the figure suggest that by increasing the total size of the dataset would lead the two curves to come closer and closer, meaning that the model would be able to generalize pretty-well.

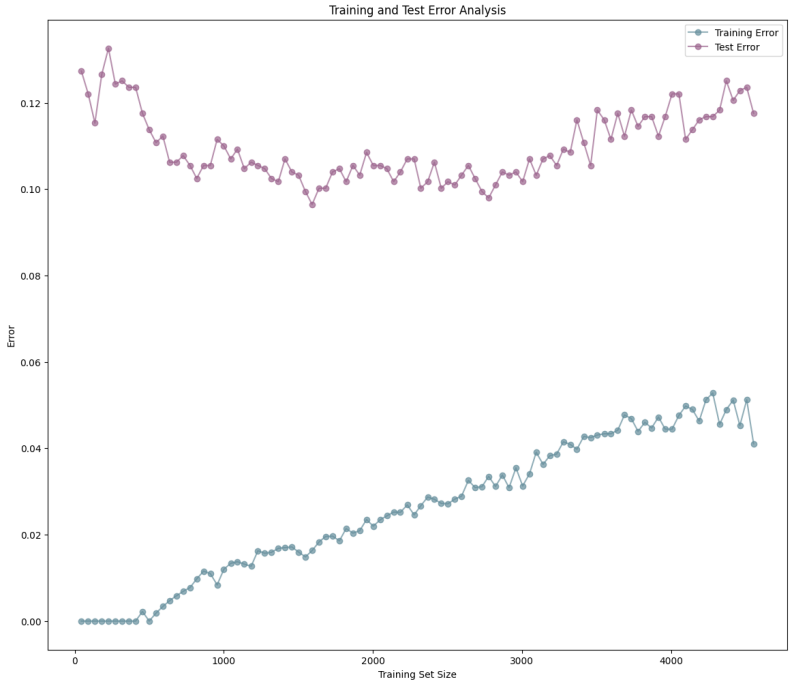


Figure 10 : Training and Test Error for Different Training Data Sizes

Final Performance Metrics

After conducting bias, variance, and generalization analysis of the model, the performance metrics can be calculated on the test set. The final model demonstrates promising predictability power, as all the obtained scores exceed **0.85**. In more detail, the calculated metrics on the unseen test set are: **Accuracy:** 0.8809, **Precision**: 0.9361, **Recall:** 0.9228 and **F1-score:** 0.9294.

The confusion matrix of Figure 11. provides a more detailed breakdown of the model's predictions. It shows the number of samples that were correctly classified as ***Alive*** or ***Dead*** and those that were misclassified. In this case, out of 1127 true ***Alive*** instances, the model correctly predicted 1040 as ***Alive*** and misclassified 87 as ***Dead***. Similarly, out of 200 true ***Dead*** instances, the model correctly predicted 129 as ***Dead*** and misclassified 71 as ***Alive***. Minimizing misclassification of *Dead* labels as *Alive* would be part of future work.

Overall, the **Random Forest** classifier demonstrated strong performance on the test data, achieving high accuracy, precision, recall and F1-score. The model's ability to correctly classify instances of the **Alive** class was particularly remarkable. However, further analysis

and consideration of domain-specific requirements are necessary to fully evaluate the suitability of the model for the intended purpose.

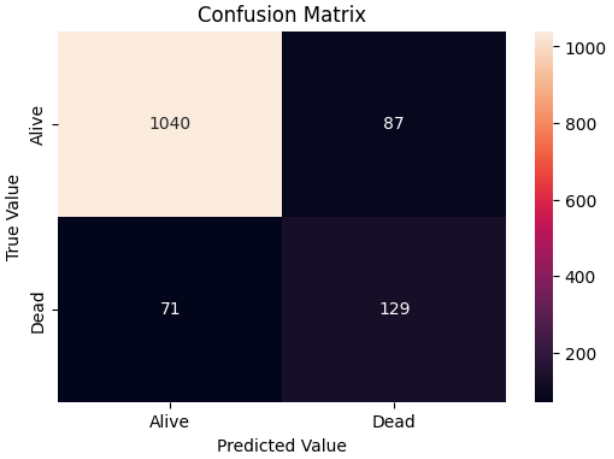


Figure 11: Confusion Matrix Calculated on the Test Data

Interpretability of the Final Model

It is crucial to have the ability to interpret a model's performance when generating future predictions. This allows us to identify and address issues such as high bias or variance and work towards achieving even better results. From the analysis of Figure 12., it is evident that the most important feature in the final model is *survival_months*. The predictive power of the model is also heavily influenced by *positive_lymph_ratio* and *age_at_diagnosis*. Additionally, *tumor_size* and *n_stage_N1* features contribute significantly to the model's ability to distinguish between *Alive* and *Dead* cases. Moreover, *n_stage_N3*, 6th_stage_ICC, and *progesterone_status_negative*, are ranked next in terms of feature importance, while the remaining variables contribute relatively less.

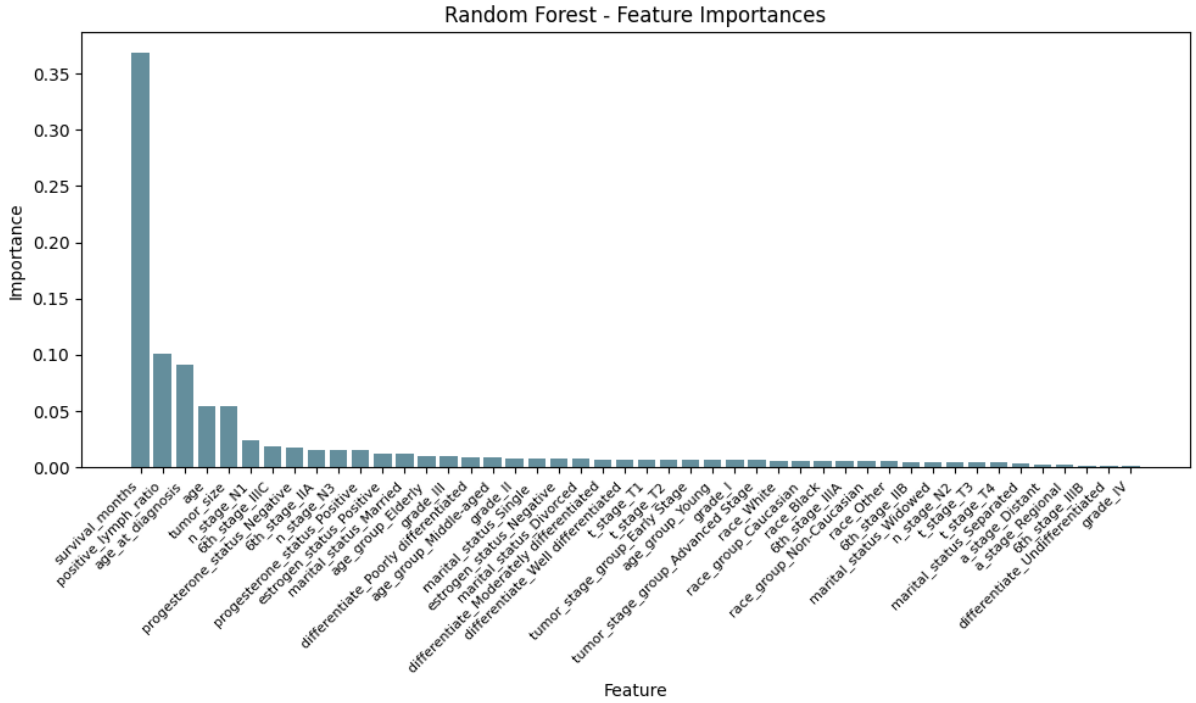


Figure 12: Feature Importance of the Final Model

Limitations & Future Work

It is important to note a couple of key observations. Firstly, there is some overlap between certain variables used in training the algorithm, such as features *Age* and *age_at_diagnosis*. Secondly, certain categories within the architecture do not significantly contribute to decision-making for unseen data. However, since the computational requirements of the algorithm, including processes like cross-validation, were manageable in this case, special efforts to reduce model dimensionality were not prioritized.

Nevertheless, dimensionality reduction techniques like *PCA/MCA/MFA* could be employed to address this issue and reduce the size and complexity of the model. Additionally, by examining the *chi-2-2 CSV* file (delivered), it is possible to identify statistically correlated categories of the categorical variables. Those associations could be minimized with the dimensionality reduction techniques mentioned in future work, as they introduce minor limitations in the current solution.

Conclusion

In conclusion, this report focused on the exploratory analysis as well as on the modeling phase of the project, encompassing, EDA, dataset splits, preprocessing, selection of learning algorithms, model comparison, hyperparameter tuning, and the final model evaluation. After the initial univariate and bivariate analyses the dataset was carefully split into training and testing sets, allowing for unbiased evaluation of the models' performance. Preprocessing steps, including normalization and one-hot encoding, were applied separately to training and test data to simulate real-case scenarios. Various learning algorithms were compared, and the Random Forest algorithm emerged as the most suitable for the specific use-case, prioritizing high recall to minimize misclassification of deceased cases. Furthermore, hyperparameter tuning was performed using cross-validation and grid search, resulting in optimal parameter values for the final model. Additionally, training-test error analyses were conducted to ensure the model's fit and generalization. Consequently, the final model demonstrated promising predictability power, achieving high accuracy, precision, recall, and F1-score on the test set. The interpretability of the model highlighted key features influencing its predictions. Limitations were identified, such as overlapping variables and categories with minimal impact, suggesting potential areas for future work, including dimensionality reduction techniques and further analysis of correlated categorical variables.