UPC UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

FIB

Breast Cancer Prediction

Machine Learning

Authors:
Bogdana Živković
Nikola Ivanović

Supervised by:
Marta Arias Vicente
Bernat Coma Puig

2023

1 Introduction

Breast cancer is a significant global health issue, affecting over 1.5 million women annually according to the World Health Organization. Existing methods for breast cancer diagnosis pose challenges such as complexity, cost, human-dependency, and inaccuracy. In recent years, computerized and interdisciplinary systems have emerged to address these limitations and minimize human errors in quantification and diagnosis. This project aims to leverage machine learning techniques to accurately determine the mortality of patients suffering from breast cancer.

For this project, we will utilize a breast cancer patient dataset obtained from the November 2017 update of the Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI). The dataset focuses on female patients diagnosed with infiltrating ductal and lobular carcinoma breast cancer diagnosed between 2006 and 2010 [1].

2 Related Work

The dataset we are using for this project has been previously employed in studies related to breast cancer survival prediction. Different methods and algorithms have been explored in order to predict mortality of patients using this dataset. Furthermore, the SEER Breast Cancer Data on the Kaggle website has attracted significant attention, with 50 code implementations submitted. These previous works provide valuable insights for our project [1], [2].

3 Data Exploration

The dataset consists of 16 columns and 4024 rows, representing various attributes of the patients. These attributes include Age, Race, Marital Status, T Stage, N Stage, 6th Stage, Differentiate, Grade, A Stage, Tumor Size, Estrogen Status, Progesterone Status, Regional Node Examined, Regional Node Positive, Survival Months, and Status. Among these attributes, five are numerical, while the remaining eleven are categorical. Importantly, the dataset contains no missing values. In the data exploration process, we aimed to gain insights into the dataset by inspecting the distributions and relationships of the variables.

We divided the dataset into numerical and categorical columns, and then using the seaborn library and pandas crosstab function, we visualized the distribution of each variable with respect to the target variable, Status, which indicates the patient's survival outcome. Numerical columns were plotted as histograms, while categorical columns were represented as stacked bar plots.

During the data exploration phase, we observed a correlation between the distributions of individual features and patients survival outcomes. This finding suggests that certain features may have a more significant influence on predicting patient survival than others. This knowledge can inform our modeling process, allowing us to focus on the most informative features and potentially improve the accuracy of our predictions.

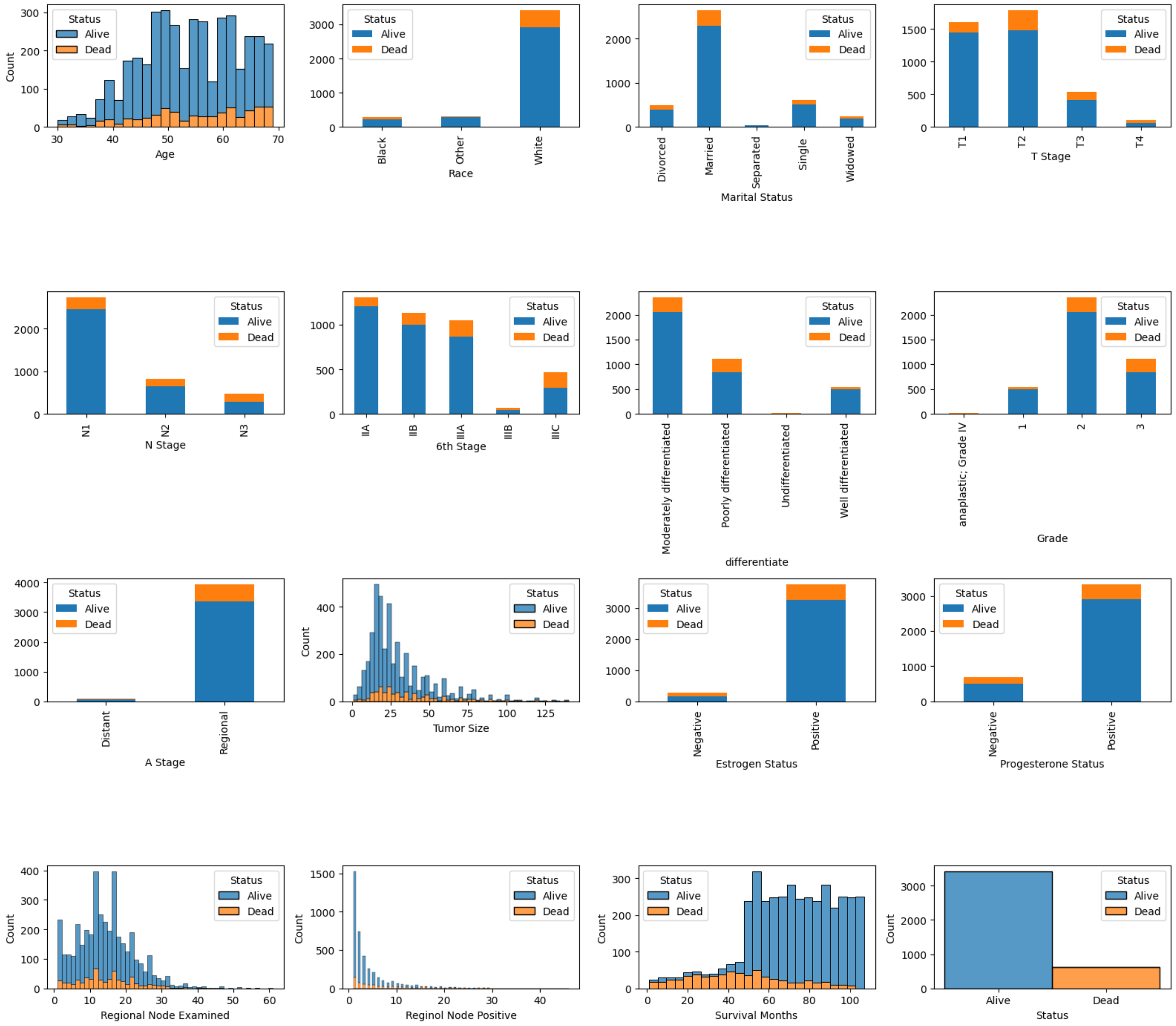


Figure 1. Correlation between individual features and patients survival outcomes

Observations derived from the presented graph indicate that the survival month feature has the greatest influence on survival outcome. Additionally, it is evident that certain categorical features, such as race and marital status, have a comparatively small role in determining patient survival outcomes. Despite this finding, we decided to keep all of the features because we didn't want to lose any information, knowing that several models that we've utilized already possess inherent feature selection capabilities.

Additionally, the numerical columns were passed to the boxplot() function, which provides a visual representation of the distribution of numerical data, showing the median, quartiles, and any outliers or extreme values.

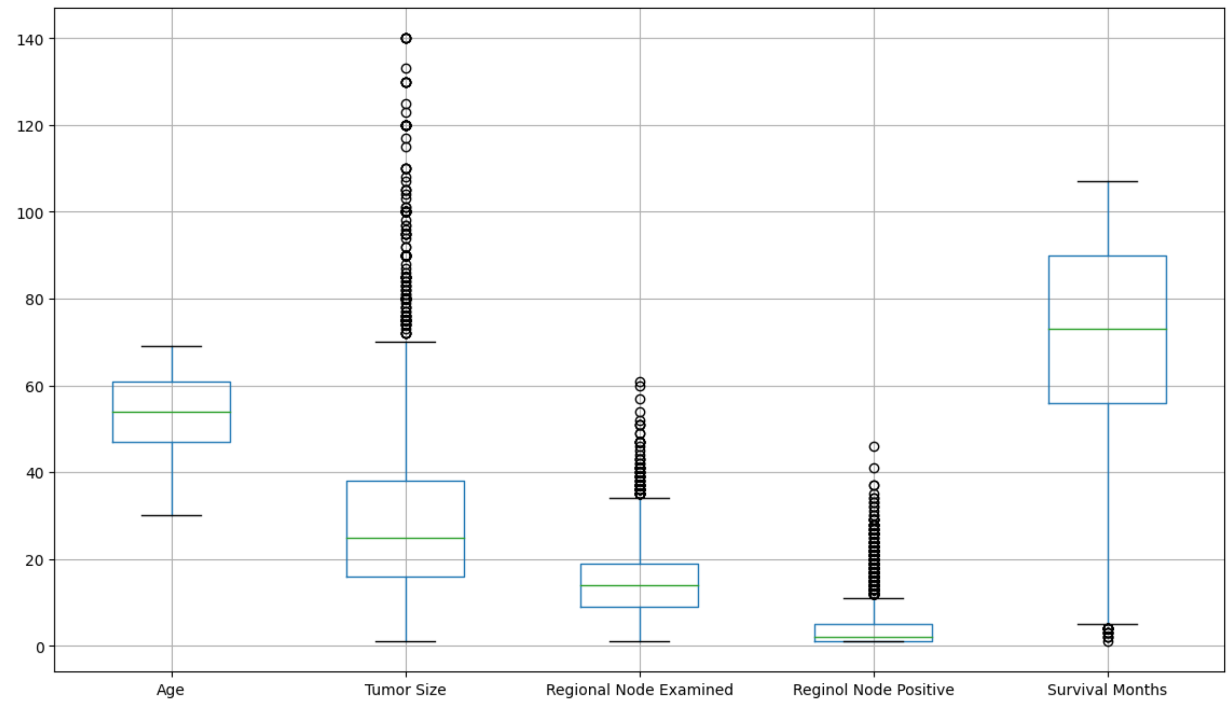


Figure 2. Distribution of numerical data

After considering both graphs, it becomes apparent that certain features demonstrate skewed normal distributions. To address skewness of data, we applied logarithmic transformations to the 'Regional Node Examined' and 'Tumor Size' columns. This was done to normalize the distributions and improve the performance of certain algorithms.

The dataset was split into training, validation, and test sets using the train_test_split function from the scikit-learn library. We allocated 80% of the data for training, while 20% was reserved for both validation and testing. Stratification was employed to ensure a proportional distribution of the target variable in each subset. By employing stratification we aimed to mitigate the risk of biased or skewed training, testing, or validation processes. This approach helps to enhance the robustness and generalization capabilities of the model by allowing it to learn and evaluate performance across various classes more effectively.

We made the deliberate choice to utilize the traditional train, test, and validation split instead of cross-validation for several reasons. Firstly, this approach afforded us the flexibility to apply data balancing techniques exclusively on the training set, enabling us to address class imbalance effectively while preserving the original distribution within the validation set. Additionally, it is worth noting that certain models employed in our analysis, such as Random Forest, inherently incorporate cross-validation as an integral part of their algorithmic framework. Considering these factors, we concluded that adopting the standard split methodology was more suitable for our specific requirements. Finally, we thought this approach was justified considering the size of the dataset which consists of 4024 rows.

Another step of data preprocessing was simple outlier detection. Outliers were detected using the interquartile range (IQR) method. For the 'Tumor Size' and 'Regional Node Examined' columns, we calculated the IQR range, defined the lower and upper bounds for outliers, and identified the outliers. We then removed the outliers from the training set, while leaving test and validation sets intact.

Next, a preprocessing function was defined to scale the numerical columns using the MinMaxScaler from scikit-learn. We also performed one-hot encoding on the categorical columns using the get_dummies function from pandas. Finally, the 'Status' column was encoded as binary values ('Dead': 1, 'Alive': 0) to facilitate model training. The preprocessing function was then applied to the training, validation, and test sets, with the scaler being fit only on the training set to prevent information leakage.

As previously mentioned, we observed an imbalance in the distribution of patient outcomes. Specifically, there were significantly more patients who survived than those who did not. This class imbalance could potentially introduce bias and affect the performance of our models. To address this issue, we applied the SMOTEENN technique, which combines oversampling (SMOTE) and undersampling (Edited Nearest Neighbors). This approach allowed us to generate synthetic samples for the minority class (patients who did not survive) while removing instances from the majority class (patients who survived). As a result, we achieved a more balanced distribution of patient outcomes in the training set, enhancing the robustness and fairness of our subsequent analyses and models.

4 Evaluation Metrics

In evaluating the performance of classification models, it is crucial to select appropriate metrics that provide insights into different aspects of the model's effectiveness. The metrics that we have chosen for evaluating model performance include:

1. Recall and F1-score for minority class,
2. Accuracy for the overall dataset,
3. F1-score, precision, and recall with macro averaging.

When dealing with a heavily imbalanced dataset, it is crucial to consider metrics that provide insights specifically about the minority class. Considering metrics specifically about the minority class is important because in our case it represents the class of interest that we want to accurately identify. In imbalanced datasets, where the majority class dominates the distribution and the minority class is underrepresented, traditional evaluation metric can be misleading. By considering metrics specifically about the minority class, such as Recall or F1-score, we focus on the model's performance in correctly identifying positive instances of the minority class. These metrics provide insights into the model's ability to minimize false negatives and are particularly relevant when the consequences of missing positive instances are significant, which is especially true in cases of medical prognosis like ours.

Additionally, Accuracy and F1-score, Precision, and Recall with macro averaging are valuable metrics for evaluating overall model performance. Accuracy represents the proportion of correctly classified instances in the entire dataset and provides a general overview of the model's predictive power. F1-score with macro averaging considers the balance between precision and recall across all classes, providing an aggregate measure of the model's performance. Precision with macro averaging measures the ability of the model to limit false positives across all classes, while Recall with macro averaging captures the model's ability to identify instances of all classes.

By including these metrics, we ensure a comprehensive evaluation of the model's performance, giving attention to the imbalanced nature of the dataset and providing insights into both the minority class and the overall predictive capabilities of the model. The presented combination of metrics allows us to evaluate the model's performance from various angles and make informed decisions based on the specific requirements and priorities of the classification task. When everything is taken into account, we selected recall of the minority class for the primary metric used to select the best models, emphasizing accurate detection of positive instances from the underrepresented class. However, the remaining metrics were important too, as general performance indicators.

In addition to the previously discussed metrics, we computed confusion matrices in order to obtain a comprehensive summary of the predicted and actual classifications made by models. They allowed us to visualize the performance of the model by presenting the number of true positives, true negatives, false positives, and false negatives.

5 Modeling Methods

5.1 LDA

One of the modeling techniques we implemented is the Linear Discriminant Analysis (LDA). LDA is a powerful method commonly used in classification tasks to find linear combinations of features that maximize the separation between classes.

We examined the performance of LDA by visualizing the transformed features. We used the transform function of the LDA model to project the training data (X_train) into a lower-dimensional space. By plotting a scatterplot of the transformed features, we gained visual insights into the separation between the two classes, as indicated by the hue encoding. This visualization helped us understand how well the LDA model was able to differentiate between the two classes based on the learned linear discriminants.

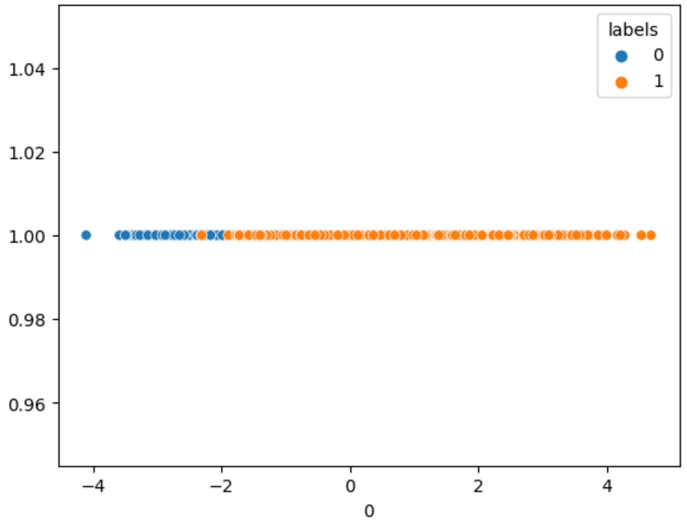


Figure 3. LDA class separation

From the results we can observe that while the class separation exists it is not perfect, which is reflected in this model's validation results.

5.2 K-Nearest Neighbors (KNN)

We also employed the K-Nearest Neighbors (KNN) algorithm as another modeling method. Using a range of k values from 1 to 50, we trained multiple KNN classifiers and evaluated their performance on the validation set. By specifying the Minkowski distance metric, the algorithm was set to calculate distances between data points using the Minkowski distance formula. The evaluation metrics were computed for each k value. The results were then sorted based on the class 1 recall metric in descending order. Among the evaluated k values, the KNN model that achieved the highest recall was selected.

5.3 Logistic Regression

Logistic regression is a commonly used modeling technique in machine learning. Its advantages include simplicity, interpretability, and efficiency. It works well with binary classification problems and is particularly useful when the relationship between the input features and the target variable is approximately linear. Our code performs logistic regression with different regularization strengths, selects the best model based on the recall of the minority class, and computes various evaluation metrics. We also generate a confusion matrix to assess the model's performance on the validation data.

5.4 Random Forest

Random forest is a powerful machine learning algorithm commonly used for binary classification problems. It is a method that combines multiple decision trees to make predictions. Each decision tree in the random forest is trained on a random subset of the training data and uses a random subset of features to split the nodes.

In our code a parameter grid was defined for the GridSearchCV function, specifying different values for hyperparameters such as the number of trees (n_estimators), maximum depth of each tree (max_depth), and minimum number of samples required to split an internal node (min_samples_split). A random forest classifier was created with a specified random state. The GridSearchCV object was instantiated with the random forest classifier, the parameter grid, and various scoring metrics. A 5-fold cross-validation was performed (cv=5), and the scoring metric chosen for refitting the best model was recall (refit='recall'). The GridSearchCV object was then fitted to the training data (X_train and y_train). The best model was determined by the highest recall score. Finally, using the best random forest model, predictions were made on the validation data.

Another useful feature of random forest models is that they allow analysis of feature importance. Performing feature importance analysis is crucial for gaining insights into the predictive power of different features and their impact on the model's performance. By evaluating feature importance, we can identify the most influential variables or attributes that have a significant effect on the target variable. This analysis allows us to prioritize and focus on the most relevant features, potentially improving the model's performance, interpretability, and efficiency. Furthermore, feature importance analysis helps in identifying potential redundancies or irrelevant features that may not contribute significantly to the model's predictive power. Removing such features can simplify the model and reduce complexity, leading to faster training and inference times.

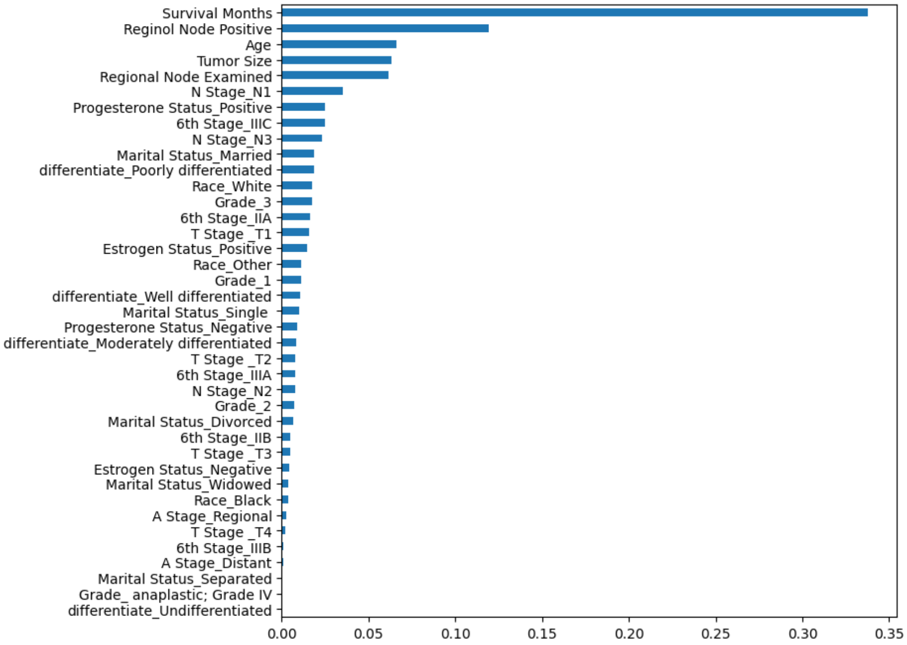


Figure 4. Random Forest feature importance

5.5 Gaussian Naive Bayes (GNB)

Upon conducting feature importance analysis, we discovered that the numerical features held greater significance in our model compared to categorical features. Building upon this insight, we proceeded to employ Gaussian Naive Bayes (GNB) as our classification technique. GNB leverages the assumption of feature independence and is well-suited for datasets with limited samples. By focusing on the influential numerical features, we aimed to enhance the accuracy and interpretability of our predictions. Additionally, GNB's probabilistic nature enabled us to estimate class probabilities, making it particularly valuable in scenarios such as medical diagnosis where risk assessment and decision-making rely on accurate probability estimates.

5.6 Neural Networks

Neural networks offer advanced modeling capabilities, adaptability to complex data, and the potential to uncover intricate relationships, making them valuable tools in medical diagnosis. We employed Neural Networks as part of our modeling approach, leveraging the MLPClassifier implementation from the scikit-learn library.

Initially, we defined a set of potential network architectures by specifying different sizes for the hidden layers. The sizes were systematically varied, starting from single-layer configurations and gradually increasing to more complex architectures with multiple hidden layers. The purpose of exploring different sizes was to capture the potential nonlinear relationships and intricate patterns within the data. Furthermore, we incorporated the concept of weight decay, represented by the 'alpha' parameter, to mitigate overfitting. By introducing regularization through weight decay, we aimed to improve the model's generalization capabilities and prevent excessive reliance on specific features. To determine the optimal architecture and weight decay, we employed GridSearchCV, a powerful tool for hyperparameter tuning. By defining a parameter grid encompassing various hidden layer sizes and weight decay values, we systematically searched for the combination that yielded the best performance across the evaluation metrics we have selected. After fitting the GridSearchCV object to the training data, we obtained the best parameters and corresponding score. The best model, identified as the one with the highest recall for class 1, was then selected and used for prediction on the validation set. The resulting predictions were evaluated using our predefined set of metrics, and the performance metrics were recorded in the results dataframe.

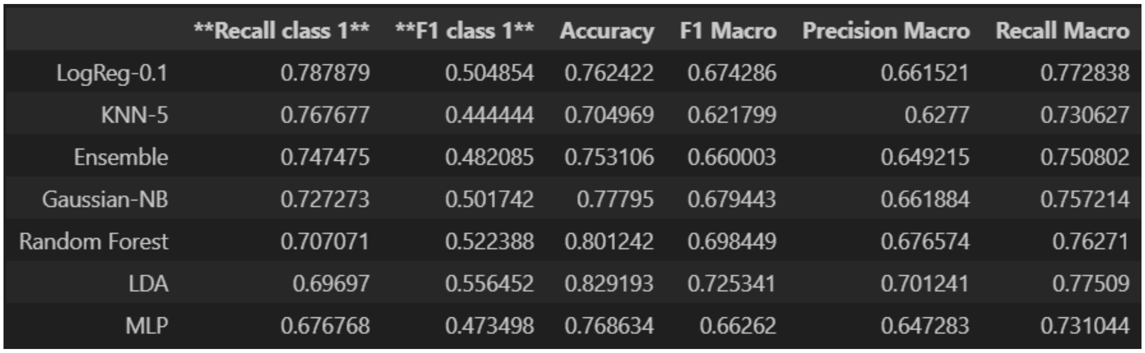
6 Ensemble

Ensembles represent combinations of multiple individual models used for making predictions or decisions. Instead of relying on a single model, ensembles leverage the collective wisdom of multiple models to improve overall performance and generalization.

We used a Voting Classifier ensemble in scikit-learn for combining the predictions of multiple individual classifiers. First, we defined the ensemble using the VotingClassifier class from scikit-learn's ensemble module. For constructing the ensemble we used the random forest model along with the two best performing models: KNN and logistic regression. By setting the voting parameter to 'soft', we employed soft voting in our ensemble approach. Unlike hard voting, where each model in the ensemble makes a discrete prediction, soft voting ensemble takes into account the predicted probabilities of each model. Soft voting combines the predicted probabilities from each classifier and selects the class with the highest average probability as the final prediction. The ensemble is then fitted to the training data, where it learns to combine the predictions of the individual classifiers. After fitting the ensemble, it is used to predict the class labels of the validation data. Finally, the performance of the ensemble is evaluated with the same metrics used for the individual models.

7 Model Selection Results

Upon analyzing the results obtained from various models, it becomes evident that both K-Nearest Neighbors (KNN) and Logistic Regression exhibit superior performance in terms of recall for the minority class. However, it is important to note that their overall performance does not surpass other models when considering all the other evaluation metrics. Even though more complex models, such as MLP and Random Forest, outperform KNN and Logistic Regression on other performance metrics, the simpler models are preferable because of the specific problem we're trying to solve. That is, the model has to be optimal for detecting true positives which is why we're prioritizing high recall scores. A high recall score reflects the model's ability to effectively capture positive instances, reducing the risk of false negatives and ensuring that fewer positive cases go undetected. Finally, we have decided to test generalization performance on Logistic Regression as it performed the best in terms of recall for the minority class.



	Recall class 1	**F1 class 1**	Accuracy	F1 Macro	Precision Macro	Recall Macro
LogReg-0.1	0.787879	0.504854	0.762422	0.674286	0.661521	0.772838
KNN-5	0.767677	0.444444	0.704969	0.621799	0.6277	0.730627
Ensemble	0.747475	0.482085	0.753106	0.660003	0.649215	0.750802
Gaussian-NB	0.727273	0.501742	0.77795	0.679443	0.661884	0.757214
Random Forest	0.707071	0.522388	0.801242	0.698449	0.676574	0.76271
LDA	0.69697	0.556452	0.829193	0.725341	0.701241	0.77509
MLP	0.676768	0.473498	0.768634	0.66262	0.647283	0.731044

Table 1. Model selection results

8 Generalization Performance

The following results represent generalization performance outputs for the trained Logistic Regression model for lambda value 0.1:

```
recall score:  0.7398373983739838
f1 score:  0.4789473684210527
accuracy score:  0.7540372670807454
f1 macro:  0.6589858793324777
precision score:  0.6478457212644494
recall macro:  0.748217819421596
```

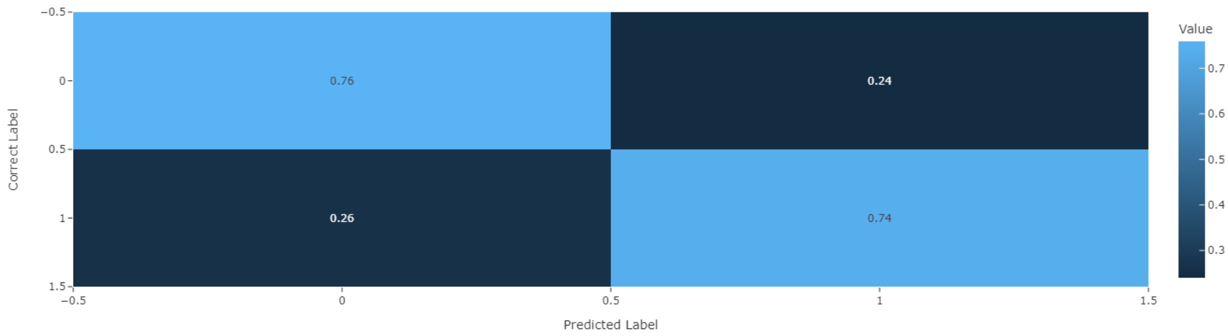


Figure 5. Generalization confusion matrix

Upon analyzing these results, several observations can be made. Firstly, the performance metric values closely resemble the ones obtained during validation, suggesting their reliability. The recall score remains relatively high, while the other scores also indicate a good general performance of the model. However, there is one exception: the F1 score for the minority class which indicates intermediate or moderate performance. The reason for this is that F1 takes into account the precision of the model (TP/(TP + FP)) which can be a misleading metric for a heavily underrepresented class, as the number of false positives can exceed the number of true positives even in the case of a well performing model. When examining the confusion matrix, we can observe that the model is equally as good at predicting the minority class as the majority class which is indicated by similar TP and FN values. Achieving this balance in predictive performance between the two classes was one of the primary objectives of our approach. This result signifies a notable achievement, as it indicates that our model effectively captures instances from the minority class without compromising its ability to predict the majority class accurately.

9 Conclusions, Possible Extensions and Known Limitations

The primary challenge we encountered revolved around the class imbalance of the target variable as the entirety of the pipeline revolved around dealing with this issue, from the selection of adequate metrics, to data balancing, to the selection of the validation protocol. Although the results suggest that this issue has been addressed to an extent, it still presented a drawback for the predictive capabilities of our model. Secondly, the dataset exhibited an interesting characteristic, as numerical features emerged as more influential than categorical ones, particularly in relation to the 'Survival Months' variable, which displayed clear dominance. This observation was initially hinted at during the dataset exploration phase but was later confirmed by analyzing feature importance with decision trees.

Despite this solution incorporating a diverse range of models and techniques, it should be noted that it represents only a subset of the extensive array of available methods. For instance, by exploring a model-based outlier detection approach and fine-tuning it, we could potentially achieve superior performance compared to the current implementation using the IQR method. Furthermore, there’s a room for more balancing or scaling algorithms to be tested in order to enhance the overall effectiveness of the solution. Finally, it is worth mentioning that certain models, such as MLP and Random Forest, could have been subjected to a more extensive validation process involving a broader range of hyperparameters.

However, in light of the project's primary objective, which focused on examining the capabilities of different models for addressing this specific issue, we made a deliberate choice not to prioritize this because of the long execution time. For this reason, certain models such as the Extra Trees Classifier weren't employed either.

In regard to the final results, it should be noted that various approaches that were not included in the final solution were unable to surpass 0.75 recall for the minority class. This limitation can be attributed to the inherent constraints of the dataset we utilized in our analysis.

10 References

[1] Namdari, Reihaneh. "Breast Cancer." Kaggle, 8 Aug. 2022, https://www.kaggle.com/datasets/reihanenamdari/breastcancer?resource=download.
[2] Haque, Mohammad Nazmul, et al. "Predicting Characteristics Associated with Breast Cancer Survival Using Multiple Machine Learning Approaches." Computational and Mathematical Methods in Medicine, vol. 2022, 2022, pp. 1–12.,https://doi.org/10.1155/2022/1249692.