Enhancing Breast Tumor Diagnosis: A Comprehensive Analysis

Data Science II

COSC 4337

Submitted to

Dr. Ricardo Vilalta

Ly Ha 1920058

Erica Hay 2049545

Khuong Ngo 1857210

In 1992, research conducted at the University of Wisconsin aimed to uncover correlations between cell characteristics and contextual features, with the ultimate goal of enhancing diagnostic accuracy by identifying traits linked to malignancy. The features analyzed describe various characteristics of cell nuclei present in the images. Leveraging machine learning techniques, our goal is to predict whether a given breast mass is benign or malignant based on these features. The importance of precise classification in breast cancer diagnosis cannot be overstated, given its pivotal role in treatment planning and patient outcomes. With a dataset consisting of 30 predictor variables, there is a likelihood of inter-variable correlation. Consequently, the selection of appropriate machine learning models becomes pivotal. Hence, in this report, we explore the effectiveness of multiple logistic regression, decision trees, and random forest classifiers in accurately classifying the binary target variable. Our objective here is to assess and compare the performance of these models in discerning the nature of breast masses with precision. Through evaluation and comparison of these models, we seek to provide insights into their relative strengths and weaknesses, aiding clinicians with valuable information for making well-informed decisions in breast cancer diagnosis and treatment planning.

Multiple logistic regression is a statistical technique used for binary classification tasks, where the goal is to predict the probability of a binary outcome based on multiple predictor variables. Given the nature of our dataset, which contains 30 predictor variables, multiple logistic regression emerges as an obvious modeling approach. Its inherent interpretability allows us to easily interpret the relationship between each predictor variable and the probability of a breast mass being malignant or benign. This interpretability is crucial in medical contexts where understanding the factors influencing disease diagnosis is paramount. Additionally, multiple logistic assumes a linear relationship between the log odds of the outcome and the predictor variables, which is often a reasonable approximation for real-world datasets like ours. Furthermore, with 30 predictor variables, there’s a likelihood of inter-variable correlation in our dataset. Multiple logistic regression can handle correlated predictors reasonably well without requiring additional preprocessing steps. Moreover, logistics regression can be regularized to prevent overfitting, a common concern in high-dimensional datasets like ours. Regularization techniques such as L1 or L2 regularization help improve the model’s generalization performance by penalizing overly complex models. Finally, logistic regression provides probabilistic outputs, enabling clinicians to interpret the model's predictions as the likelihood of a breast mass being malignant or benign. This probabilistic nature enhances the model's transparency and facilitates decision-making in clinical practice.

We proceed to fit the multiple logistic regression model to our dataset. This pivotal step demands careful consideration and tuning of several hyperparameters to attain optimal performance. One crucial hyperparameter is the penalty parameter, which determines the type of regularization applied to the model. This parameter can take values such as “l1” for L1 regularization (Lasso) or “l2” for L2 regularization (Ridge). Additionally, the C parameter plays a significant role in controlling the strength of regularization. Representing the inverse of the regularization strength, smaller C values enforce stronger regularization, while larger values lead to less regularization, allowing the model to closely fit the training data. Balancing the regularization strength with the complexity of the model is essential for achieving a well-generalized and robust logistic regression model. To accelerate the process of finding the optimal model configuration, we utilized a method called “GridSearchCV”. This method systematically explores a predefined grid of hyperparameters and evaluates the model's performance using cross-validation. Our grid specification encompassed two sets of hyperparameters: one for the penalty parameter (“l1” and “l2”) and another for the C parameter (0.001, 0.01, 0.1, 1, 10, 100, 1000). Employing 5-fold cross-validation, the GridSearchCV method meticulously iterates through each combination of hyperparameters. For every combination, it trained the model on a subset of the training data (training fold) and evaluated its performance on a distinct subset (validation fold). With 14 hyperparameter combinations fitted across 5 folds each, totaling 70 iterations, GridSearchCV calculated performance metrics such as accuracy, precision, recall, or F1-score for each validation fold. Our choice of performance metric was accuracy.

Upon assessing all hyperparameter combinations, GridSearchCV identified the set yielding the best performance metric across all folds. Ultimately, the combination of a C value of 0.1 with L2 regularization was deemed the optimal configuration for our logistic regression model. With this “best model”, we finally delve into understanding the impact of each predictor variable on the likelihood of malignancy by examining the coefficients (**Figure #**) derived from the model. Positive coefficients indicate that an increase in the corresponding predictor variable is associated with an increase in the log odds of malignancy, while negative coefficients suggest the opposite. Additionally, the magnitude of the coefficients further explains the strength of the relationship between each predictor variable and the log odds of ‘malignant’, aiding in prioritizing features based on their impact. For instance, features such as concave\_points\_mean, radius\_se, radius\_worst, texture\_worst, perimeter\_worst, area\_worst, concavity\_worst, concave\_points\_worst, and symmetry\_worst exhibit positive coefficients with large magnitudes, implying that higher values of these features contribute to an increased possibility of a malignant tumor. Conversely, features that exhibit negative coefficients with large magnitudes, such as fractal\_dimension\_mean, compactness\_se, symmetry\_se, and fractal\_dimension\_se, suggest that lower values of these features are associated with a decreased probability of malignancy. These odds ratios implicitly reflect the impact of each predictor variable similar to the coefficients. Specifically, for each one-unit increase of a predictor variable, the odds of malignancy are increased by the odds ratio of the respective predictor variable.

In evaluating the performance of our logistic regression model, we employed several key performance metrics: the confusion matrix, learning curves, and cross-validation. The confusion matrix (**Figure #**) revealed insightful results, indicating 46 true positives, 0 false positives, 1 false negative, and 22 true negatives. This breakdown provided a clear understanding of the model's classification performance. Further analysis involved calculating additional performance metrics: accuracy, precision, recall, and F1-score. The model demonstrated exceptional accuracy at 98.6%, signifying the proportion of correctly classified instances among all instances. Precision, measuring the proportion of true positive predictions among all positive predictions, was 100%, reflecting the low false positive rate. Recall, which assesses the proportion of true positives identified correctly, stood at 95.7%, indicating the model's ability to capture the “majority of positive instances. The F1-score reached 97.8%, affirming the model's balance between precision and recall. Moving on to learning curves (**Figure #**), we observed consistent improvement in both training and test scores as the number of training examples increased. Notably, the training score plateaued around 0.98, suggesting the model's ability to learn from additional data. Conversely, the test score exhibited slight fluctuations but remained high, indicating the model's generalization capability. Finally, through 5-fold cross-validation (**Figure #**), we obtained mean scores for accuracy, precision, and recall, providing robust estimates of the model's performance. The small standard deviations accompanying these metrics indicated minimal variability, further reinforcing the model's stability and reliability.

A screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure # and Figure #

A diagram of a logistic confusion matrix

Description automatically generatedA graph with a line and a line graph

Description automatically generated with medium confidence

Figure # and Figure #

Decision trees are used for both classification and regression tasks. It is a supervised learning algorithm that works by recursively partitioning the input space into smaller regions based on the values of input features. At each step, it selects the feature that best splits the data into its purest possible subsets in terms of the target variable. Since our goal is to see which features lead to a malignant or benign, the way the decision trees can capture nonlinear relationships between features and the target variable is ideal for our classification problem. The dataset may contain complex interactions between various features indicative of the presence or absence of cancer, which can be handled effectively with decision trees. The model can also provide insights into the hierarchy of risk factors associated with breast cancer. By analyzing decision paths, we can gain insights on the factors that contribute the most to the likelihood of a cancer diagnosis.

Moving on, let’s begin hyperparameter tuning. For the decision tree model, we went with a grid search in order to search through different options of the 5 main parameters. First, we have the `criterion` parameter which determines the function to measure the quality of a split. For the criterion, we searched through the choices of ‘gini’ or ‘entropy’. The grid search revealed that ‘gini’, which measures the impurity of the nodes based on the Gini impurity criterion, was the best for our dataset. Second, for the `max\_depth` of the tree we created a search through a range of 2 to 32. The deeper a tree, the more complex patterns it can capture, however, this also opens it to be more prone to overfitting. Thus, making it very important to control the maximum depth of the tree as it helps to prevent this, in turn increasing the performance and robustness of the model. Through the search, we landed on a max\_depth of 6. Third, we searched through a range of 1 to 9 for the `min\_samples\_leaf ` parameter. This parameter specifies the minimum number of samples required to be at a leaf node and it helps to prevent overfitting by simplifying the model the higher the value. For our dataset, this model has the best accuracy at 1 minimum samples. Next, we searched another range of 2 to 9 for the `min\_samples\_split`. This parameter is similar to that of the previous one but instead specifies the minimum number of samples required to split an internal node. It helps with overfitting by encouraging the tree to only make splits when necessary. The most optimal split was at 2 nodes. Lastly, we have the `splitter` parameter. There are two choices for this parameter, ‘best’ or ‘random’, and it determines how the algorithm selects features to split on at each node. The most optimal ‘splitter’ was the ‘random’ split which, like its name states, chooses the best random split. Overall, the best combination of parameters for our model turned out to be: {'criterion': 'gini', 'max\_depth': 6, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'splitter': 'random'}. When testing the best mean cross-validated score achieved during the search, this optimal combination gave the best score with a 0.958 or 96%.

At this point, we can now build our model. We did this through the DecisionTreeClassifier function from the sklearn.tree library. With the optimal parameters chosen previously, we got the decision tree seen in **Figure #**. In order to see how the decision tree model did performance wise, we ran the same metrics as before and we also looked at accuracy, precision, and recall.

A diagram of a company structure

Description automatically generated

*Figure #: Resulted Decision Tree from Optimal Hyperparameter GridSearch*

We used the trained decision tree classifier to make predictions on the test set and stored them into a prediction variable. Then, we calculated the accuracy of the model on both the training and test sets. For the training set, we got an accuracy of 0.988 while the testing set received an accuracy of 0.956. What does this tell us? Our model does a very good job at fitting the training data as well as provides high accuracy on unseen data. Moving on, the confusion matrix (**Figure #**) shows us that there are 44 true negatives, 2 false positives, 1 false negatives, and 22 true positives. This means that the model did very well in classifying both benign and malignant tumors. When looking at the classification report (**Figure #**), we can see that the benign class has a slightly higher precision, recall, and F1-score. This indicates that the model performs slightly better at correctly identifying these cases in comparison to malignant tumors. Moreover, the weighted average precision, recall, and F1-score is approximately 0.95-0.96, meaning that the model’s overall performance is balanced for both classes. We achieved a high precision and recall for both classes which means that the model has high predictive capabilities.A blue squares with white text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

*Figure #: Confusion Matrix for Decision Tree Figure #: Classification Report*

The next metric we evaluated were cross-validation scores. For this evaluation we got a range from 0.92 to 0.97 and the mean cross-validation score is 0.9359. This indicates that on average, the model achieves about 93.6% accuracy across the 5 different folds. Lastly, we observed the learning curve of the tree (**Figure #**). Initially, we can see that as the training set size increases the training score consistently remains high with a slight decline, ranging from 0.98 to 1.0. This suggests that the decision tree model is able to learn from the training data effectively and achieves high accuracy on the data it was trained on. Whereas there is a slight decrease in the training score trend, the cross-validation score sees an increase as the training set size increases leading to improved model performance. This suggests that the model benefits from additional training data. As we can observe, there is a significant gap in between the training score and cross-validation score lines, meaning that there is overfitting of the model. However, this area becomes more narrow as the size of the training set increases which is a positive indication as it means that overfitting is mitigated with more training data and the performance improves.

A graph showing the results of a training course

Description automatically generated

*Figure #: Learning Curve for Decision Tree*

It is very important to note that due to the random nature of the decision tree, each iteration of building the tree even with the same parameters may lead to a change in the accuracy of the model. This is due to the splitter function which randomly splits the nodes.

Random forests is a powerful learning method which builds upon the decision tree algorithm. It excels in both classification and regression tasks by aggregating the predictions of multiple decision trees, effectively reducing overfitting, increasing accuracy, and handling complex data. The random forest algorithm builds decision trees during the training phase, where each tree is made using a random subset of the training data and a random subset of the dataset’s features. This introduces variability amongst the trees, helping to reduce the risk of overfitting the data. The output of each tree is combined, and may lead to a more accurate and stable prediction compared to individual trees. The random forest algorithm can be effective in determining whether breast cancer is malignant or benign by leveraging its strengths. Breast cancer diagnosis often involves analyzing multiple complex factors, and random forests excel at handling complex interactions between the features and the target variable, allowing them to identify subtle patterns that may not be apparent using other models. Random forests can also utilize feature importance to help identify which features are most influential in determining whether a tumor is malignant or benign. Overall, random forests offer a powerful approach to breast cancer diagnosis.

Random forests contain multiple hyperparameters that can significantly impact its performance.  Finding the optimal combination of these parameters is extremely important in fully utilizing the capabilities of the random forest algorithm. Some of the parameters are similar to those of the decision tree, such as the ‘criterion’, ‘max\_depth’, ‘min\_samples\_split’, and ‘min\_samples\_leaf’. The ‘criterion’ parameter also searches through the options of ‘gini’ or ‘entropy’, the ‘max\_depth’ is set to a range of 1 to 11, and both ‘min\_samples\_split’ and ‘min\_samples\_leaf’ are set to a range of 2 to 20. However, random forests exhibit one dissimilar parameter, ‘n\_estimators’, which is the number of trees in the forest. Increasing the number of trees improves performance, but increases computational cost, so we set the ‘n\_estimators’ parameter from 10 to 400 to balance the performance and cost. After setting these parameters, we employed Bayesian optimization using the ‘BayesSearchCV’ function to tune the hyperparameters of the random forest classifier. With ‘n\_iter = 100’ and ‘cv = 5’, the search explored 100 iterations within the hyperparameter space while specifying 5-fold cross-validation to evaluate the performance of each combination. We then fit this to the training data and obtained the best combination of the parameters using the ‘best\_estimator’ function: {criterion='entropy', max\_depth=8, min\_samples\_leaf = 2, n\_estimators=251}.

After completing the hyperparameter tuning for the random forest and using the best model to perform predictions on the test set, we can further evaluate the model’s performance. First, we calculate the accuracy of the model. For the training set, we calculated an accuracy score of 99.60%, and we calculated an accuracy score of 97.10% for the testing set, indicating that the model has learned the training set well and that the model is performing well on unseen data. In addition to accuracy calculations, we can also assess the precision, recall, and F1-scores to gain a better understanding of how effectively the random forest classifies breast cancer as malignant or benign **(Figure #)**. For the benign class (0), the precision, recall, and F1-scores all have a high score of 0.98, while the malignant class (1) has a high score of 0.96 for each. This tells us that the model performs very well on classifying both benign and malignant cases, but classifies benign cases slightly better. We then plot the confusion matrix **(Figure #)** to visualize the model’s performance. The model correctly predicted 45 instances of malignant tumors and 22 instances of benign tumors, and incorrectly predicted 1 instance as benign when it was actually malignant and 1 instance as malignant when it was actually benign. Overall, the confusion matrix shows that the model performed excellently when tasked with classifying the tumors.

A blue squares with white text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

*Figure #: Confusion Matrix for Random Forest Figure #: Classification Report*

Next, we evaluated the cross validation scores for the random forest model. The scores range from 0.9386 to 0.9912, and have a mean score of 0.9683. This tells us that the model performed consistently well across the different folds of the data, with an average accuracy of about 96.83%. Then we observed the learning curve **(Figure #)** to visualize how the model’s performance changes as the training data size increases. The cross-validation scores start at 0.70 experiences a steep increase to about 0.94 at approximately 150 training examples, and then steadily increases to 0.96. The training scores do not experience any fluctuations or steep changes, starting at 1.00 and slightly decreasing to 0.99. We can see that the model stabilizes at around 240 training examples, indicating that the model may have reached its optimal performance at this point. Additionally, the gap between the two lines on **Figure #** was wide before 100 training examples, but became more narrow as the training examples increased suggesting that the model’s generalization performance improved with more data, and that there were no signs of overfitting.

*A graph with a green line and red line

Description automatically generated*

*Figure #: Learning Curve for Random Forest*

Finally, we observe the feature importance (**Figure #**) of the random forest model. Feature importance gives us insight on the relative contribution of each feature of the dataset to the model’s decision making process. This may help researchers understand which features are most influential in predicting whether breast cancer is malignant or benign. In our analysis, we found that the perimeter\_worst, area\_worst, concave\_points\_worst, and concave\_points\_mean features had the four highest importance scores, with values shown in **Figure #**. These insights help us understand which aspects of the data are most relevant for classifying breast tumors. A graph of blue bars with white text

Description automatically generated

A computer screen shot of a number

Description automatically generated

*Figure #: Top Ten important features for Random Forest*

After constructing each model, it is essential to conduct a comparative analysis of their performance. This enables us to discern the strengths and weaknesses of each model, aiding in the selection of the most suitable one for tumor classification. When comparing the accuracy, precision, recall, and F1-scores of each model, it is evident that there are many similarities among the results. Each model demonstrates exceptional performance, with all scores surpassing 0.92. A consistent trend across all models is that the scores for the benign class are higher than those for the malignant class. However, there is a dissimilarity between the variability in scores for each model. For the random forest model, there were not many differences between the scores, with each score being 0.98 for the benign class and 0.96 for the malignant class. The decision tree model exhibits more variability in the scores, where the values range from 0.92 to 0.98. Another difference between the models is the exceptional performance of the logistic model, which has a precision score of 1.

In addition to comparing the performance scores of each model, it is essential to compare each confusion matrix to gain insight on how each model correctly and incorrectly classified instances for each class. The confusion matrix for the multiple logistic regression model misidentified 1 tumor, while the decision tree model and the random forest model misidentified 3 and 2 tumors, respectively. This indicates that the multiple logistic regression model outperformed the decision tree and random forest models at accurately identifying instances.

Comparing the cross-validation scores for each classifier allows us to assess their performance across different folds of the dataset. We observe that the multiple logistic regression model has a mean cross-validation score of 0.98, while the decision tree and random forest models attained mean cross-validation scores of 0.9359 and 0.9683, respectively. These high mean values indicate that each model performed consistently across the five folds, but that the multiple logistic regression classifier performed slightly better among the three during cross validation. We can also see this in when comparing the learning curve plots for each model. As stated before, the learning curve for the multiple logistic plateaued at a value of 0.98, while the decision tree and random forest learning curves reached maximum values of about 0.95 and 0.97. The learning curve for multiple logistic regression classifier also exhibits a much smaller gap between the training score and cross-validation score curves compared to the decision tree and random forest models, indicating that the model was less-prone to overfitting.

Considering the performance metrics, confusion matrix analysis, cross-validation scores, and learning curve analysis, it appears that the logistic regression model performed best among the tree models for tumor classification. While each model performed well, the logistic regression model exhibited exceptional performance, indicated by its high accuracy and stability. These findings underscore the potential of logistic regression models in improving cancer diagnosis.