Application of Machine Learning Methods to Breast Tumor Diagnosis

Dianne Laboy & Camila Valdebenito

Abstract

Breast cancer remains a leading cause of morbidity and mortality among women. Determining the correct tumor type (malignant or benign) is a critical step in the diagnosis process. In the last two decades, an enormous amount of research has been devoted to the improvement of cancer diagnostics and therapeutics. Following the advances, a vast amount of data has been generated allowing for the application of computational tools and machine learning methods as part of the diagnostic process. In this study, we set out to identify which machine learning classifier performed best at classifying tumor samples from the Wisconsin Breast Cancer Data Set. We compared the performance of five supervised learning classifiers as measured by their recall score calculated using Python's library for machine learning, scikit-Learn. The classifiers tested in this project are: Logistic Regression, Decision Trees, Support Vector Machines, K-Nearest Neighbor, and Linear Discriminant Analysis. Preliminary data from this study suggests that Logistic Regression performed best at classifying tumor samples. We demonstrate the potential application of machine learning methods to aid physicians and clinical labs in breast cancer diagnosis.

Introduction

According to the World Health Organization, not only is breast cancer the most common cancer among women but it also inflicts the greatest number of cancer-related deaths. Early and correct diagnosis is key to ensure timely treatment and increase the likelihood of survival. Even with the increase of biomedical advancements to diagnose and treat cancer, a large proportion of breast cancers are still misdiagnosed. One of the major hurdles in the diagnosis process is determining whether a breast mass is benign or malignant.

Previous research has outlined the possibilities of applying machine learning methods in the diagnosis process (Kononenko, 2001 & Kourou et al., 2014). Machine learning tools have the ability to detect key features and patterns in complex datasets which otherwise could have been missed (Korou et al., 2014). A machine learning approach to diagnosis has the potential to speed up and improve the overall tumor diagnostics process. In this research project, we sought out to find a machine learning classifier which could be used to determine tumor malignancy. The goal is to identify the best suited machine learning classifier to classify a tumor as malignant or benign based on phenotypic data from breast tumor samples.

The data used to test classifiers is the Wisconsin Diagnostic Breast Cancer (WDBC) data set, originally collected from researchers at the University of Wisconsin, Madison. The data set contains information gathered from fine needle aspirates' digitized images of breast tumors. Fine needle aspirate is a type of biopsy which allows for the collection of cells from a cell mass. The data set contains 569 breast tumor samples of which 212 are malignant and 357 are benign. In addition to the classification, the data contains ten real-valued features for each cell nucleus: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The researchers computed the mean, standard error and largest value of the 10 features for each image, resulting in 30 total features. We used the reported features to select five machine learning classifiers and determine the classifier which performed the best at classifying a tumor sample as malignant or benign.

Methods

1. Exploratory Data Analysis

In this section, we report the analysis performed to describe our dataset. Through this analysis, we wanted to get a broad understanding of our data by identifying any patterns, trends, outliers, mistakes, or unexpected results. As a result of this step, we gained insights about appropriate models to use and determined relationships among the explanatory variables and between explanatory and outcome variables.

a) **Data Schema:** By exploring the data schema, our goal was to identify the data types for features and diagnosis, including respective possibles values (for categorical variables) and distribution (for numerical variables). In Python, Pandas provides *dtype* mapping function, which returns the data type of each column of the selected data frame.
b) **Missing values:** We wanted to get a sense of possible missing values. Having this information would allow us to decide how we wanted to handle them before performing any analysis. In Python, Pandas provides *isnull* function, which detects missing values (NaN in numeric arrays, None/NaN in object arrays).
c) **Scatterplot Matrix:** A scatter plot matrix is table of scatter plots. For our project, having multiples scatterplots in a same page allowed us to quickly identify possible correlations between our variables. In Python, seaborn seaborn library provides the *pairplot* function to create scatterplot matrices.
d) **Correlation Matrix Heatmap:** A correlation matrix is a table showing correlation coefficients between variables. Each coefficient shows the correlation between two variables. We used a correlation matrix to identify possible correlation between our features. In Python, seaborn seaborn library provides the *heatmap* function to easily create heatmaps.

2. Testing and Training Data

To evaluate our project, we randomly split the dataset into testing and training. The training dataset was used to fit the selected classifiers, while providing visibility of each tumor sample diagnosis . After the classifier was trained, data from the testing dataset (unseen data) was used to predict the diagnosis to later compare with the actual tumor diagnosis.
For our project, we chose the test set to be composed of 30% of our total dataset. To do the splitting we utilized the *train_test_split* function provided by the *sklearn.model_selection* library.

3. Measuring Performance of Machine Learning Classifiers

In this section, we describe the different metrics used in our project to measure the performance of the selected classifiers. Since we are particularly interested in detecting malignant tumors to help patients receive treatment, we primarily focused on finding the best classifier to detect malignant tumors. Having said that, we also considered lowering the incorrect classification of benign tumors to avoid unnecessary concerns for the patients.

Confusion Matrix

The confusion matrix is a matrix that reports the counts of the True positive (TP), True negative (TN), False positive (FP), and False negative (FN) predictions of the selected classifier (see Figure 1). The confusion matrix calculates the number of correct and incorrect predictions, which are further

summarized by the number of count values and breakdown into each class. We used *sklearn.metrics* to access the confusion_matrix function to compute confusion matrices.

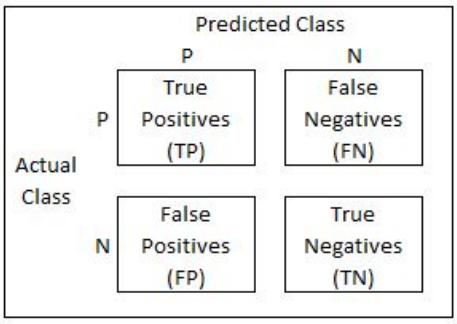


Figure 1. Confusion Matrix

In addition to the confusion matrix, we created a classification report visualizer for each of our classifiers that displays the precision, recall, F1, and support scores of each model.

a) Accuracy (A): Accuracy for the classifier corresponds to the sum of the correct predictions divided by the total number of predictions as it is showed in the formula below:

$$A = \frac{TP+TN}{TP+TN+FP+FN}$$

b) Precision (P): Precision corresponds to the proportion of positive predictions that are actually correct, and calculated as follows:

$$P = \frac{TP}{TP+FP}$$

c) Recall (R): Recall is calculated as the proportion of correct observations that are identified as positive. This is mathematically represented as follows:

$$R = \frac{TP}{TP+FN}$$

d) F1 score: The F1 score conveys the balance between the precision and the recall. This is represented by the formula below:

$$F_1 = 2\frac{PR}{P+R}$$

Accuracy, Precision, Recall and F1 core are all implemented in *scikit-learn* and can be imported from the *sklearn.metrics* module as it is shown below:

>>> from sklearn.metrics import classification_report

Given the medical importance of our project (tumor diagnosis), we believe that False Negatives are probably worse than False Positives, as this could imply that patients with malignant tumors would not receive proper treatment. From this perspective, Recall (R) would be the metric that better would represent the goal for our analysis.

4. Machine Learning Classifiers

In this section, we describe the various machine learning algorithms used to train our dataset. Given the nature of our dataset and our question, we focused on supervised machine learning classifiers to predict the diagnosis of the tumor sample. In particular, we focused on measuring the performance of five algorithms: Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbor and Linear

Discriminant Analysis. To evaluate each classifier we imported and used the corresponding function in *scikit-Learn* (sklearn) library, a python library for machine learning.

a) Logistic Regression

Logistic regression (LR) is broadly used to solve classification problems, where the independent variables are continuous in nature and dependent variable is in categorical form. Some advantages of the LR include the efficient implementation in existing libraries and provided probabilities for the outcomes. Among the disadvantages, this algorithm is known to not perform well when the feature space is too large and not be able to handle large number of categorical features. For our project, we believe main risks could be related with overfitting as our dataset contains a significantly large number of features(30).

b) Decision Tree:

Decision tree (DT) is another supervised learning algorithm used to solve regression and classification problems. This algorithm is structured just like a tree, where nodes represent features, branches represent decisions and leaves represent an outcome (in our case, benign or malignant). Among the advantages of DT we find that features that depend on each other (multicollinearity) will not affect the quality of the model. Additionally, DT are easy to interpret visually and they deal well with noisy or incomplete data. Some disadvantages are that DT are prone to overfitting and hence they tend to be unstable when introducing unseen data. Again, we believe main risks could be related with overfitting although seems like it should handle our multicollinearity well.

c) K-Nearest Neighbor

K-Nearest Neighbor (KNN) is one of the simplest supervised machine learning algorithms used for classification. KNN assigns a class to a new data point according to the classes of the data points around it. The number of data points the algorithm uses to assign a class to a data point corresponds to its n-number of neighbors. The number of neighbors can be chosen according to a value that increases accuracy of KNN. In selecting the number of neighbors it is good practice to avoid multiples of the number of classes and choose an odd number for a 2 class problems. A drawback of KNN is the complexity of searching for the nearest neighbors of a sample in a big data set. For our project, a key step is determining the number of nearest neighbor that increases the accuracy of the classifier.

d) Support Vector Machines

Support Vector Machines (SVM) is a powerful supervised machine learning method. SVM is highly effective in high dimensional spaces. SVM is efficient in creating non linear boundaries between classes by changing kernels. SVM is more complex parameters of the rbf kernel most be thoughtfully picked. SVM is ideal to classify tumor samples as with 30 features it is very unlikely a linear boundary will separate the classes. In addition, tumor samples tend to be heterogeneous which is seen in the variation for each feature in the data set, SVM might be successful at creating a boundary with this type of data.

e) Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a classic linear classifier used to model difference among groups. The scikit-learn function can extrapolates the main principle of LDA on many variables allowing for a multivariate analysis. A drawback of using LDA is that it is only able to create a linear boundary, as its name respectively states.

Results

Table 1: Machine Learning Classifiers Results summary. This table summarizes the precision, recall, and f1-score for each machine learning method tested.

Machine Learning Classifier		Precision	Recall	F1-score
LR	Training Data	B: 0.96 M: 0.95	B: 0.97 M: 0.93	B: 0.97 M: 0.94
	Test Data	B: 0.99 M: 0.96	B: 0.95 M: 0.98	B: 0.97 M: 0.95
DT	Training Data	B: 1.00 M: 1.00	B:1.00, M: 1.00	B: 1.00 M: 1.00
	Test Data	B: 0.95 M: 0.85	B: 0.91 M: 0.92	B: 0.93 M: 0.89
KNN (n=11)	Training Data	B: 0.93 M: 0.94	B: 0.96, M: 0.89	B: 0.95 M: 0.91
	Test Data	B: 0.96 M: 0.97	B: 0.98 M: 0.94	B:0.97 M: 0.95
SVM	Training Data	B: 0.95 M: 0.94	B: 0.96, M: 0.91	B: 0.96 M: 0.93
	Test Data	B: 0.96 M: 0.95	B: 0.97 M: 0.94	B: 0.97 M: 0.94
LDA	Training Data	B: 0.95 M: 0.99	B: 1.00 M: 0.91	B: 0.97 M: 0.95
	Test Data	B: 0.96 M: 1.00	B: 1.00 M: 0.92	B: 0.98 M: 0.96

Logistic Regression

In terms of F1-score, LR performed worst for malignant predictions than for benign with an F1-score of 0.95 and 0.97 respectively. In terms of precision, LR also had a lower performance for malignant predictions than for benign with a precision score of 0.96 and 0.99 respectively. Finally, in terms of recall, LR performance was better for malignant predictions than for benign with a recall score of 0.98 and 0.95 respectively.

Given the medical context of the problem, we defined our main evaluation metric to be 'recall' as it would guarantee that the chosen model would accurately classify malignant tumors as so; consequently, ensuring timely treatment of the patient to increase the likelihood of survival. In this context, LR seems to provide a good solution to the classification of malignant tumors.

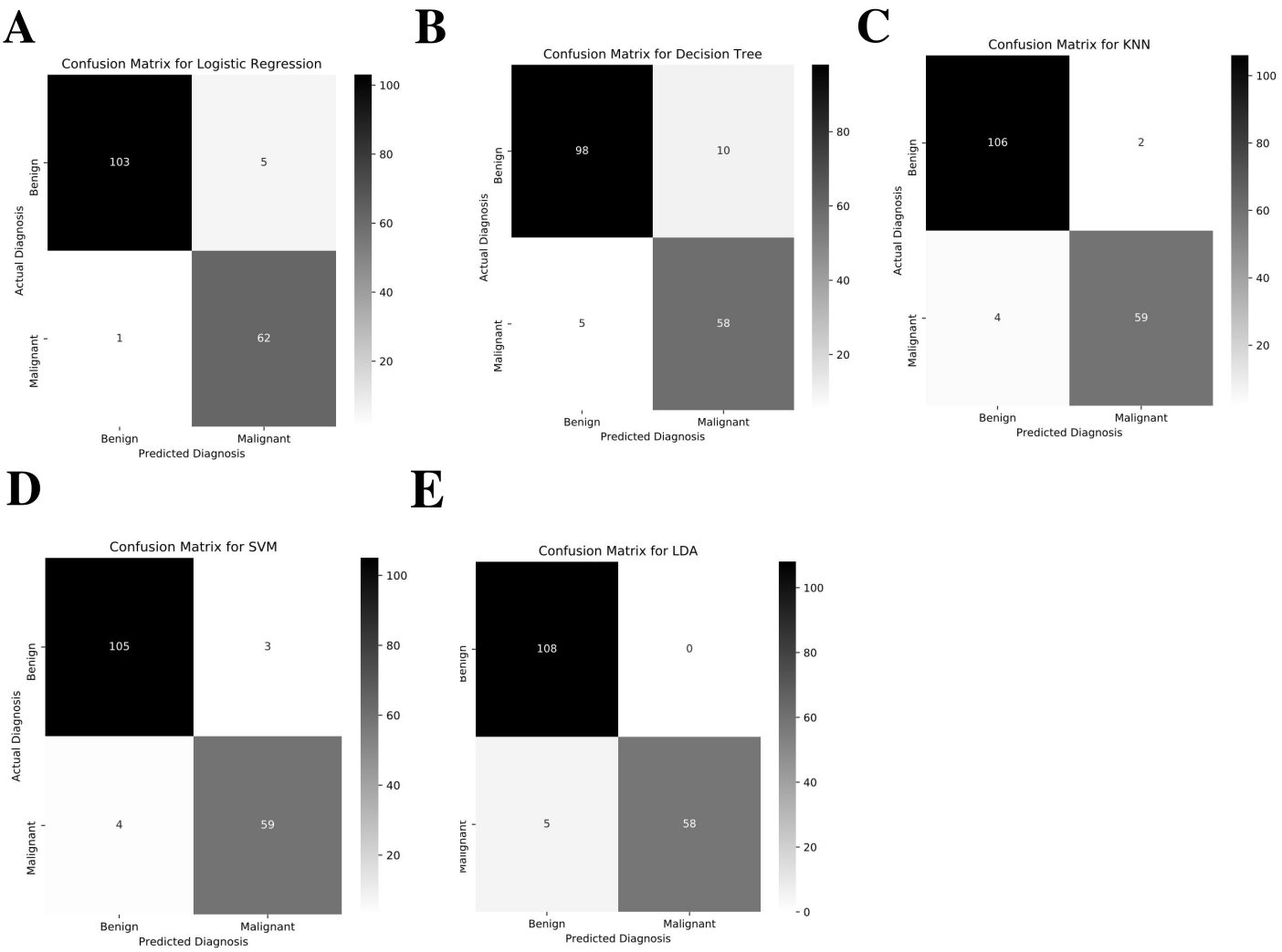
Decision Tree

In terms of F1-score, DT performance was lower for malignant predictions than for benign with a score of 0.89 and 0.93 respectively. In terms of precision, DT also had a lower performance for malignant predictions than for benign with a precision score of 0.85 and 0.95 respectively. Finally, in terms of recall, DT performance was better for malignant predictions than for benign with a recall score of 0.92 and 0.91 respectively. Even though this model does not provide high precision and F1-scores for malignant tumors, it does seem to provide high recall scores.

Given the tendency of DT to overfit, we computed the F1-score for training data as well. This analysis gave us perfect scores (1.0) for all the metrics, for both benign and malignant. Since our model shows a better fit on train samples than the test sample, we believe there was overfitting. Our results show that DT

does not seem to provide a good solution to the classification of malignant tumors. Further cross-validation analysis could be done to verify whether this is the case.

Figure 2: Confusion Matrix for each Machine Learning Classifier. A) Logistic Regression, B) Decision Tree, C) KNN (n=11), D) SVM, E) LDA. Each confusion matrix reports the counts of the True positive, True negative , False positive, and False negative predictions of the classifier using the test data.



K-Nearest Neighbor

First, a scatterplot showing the accuracy of the classifier on the test dataset for a range of n-nearest neighbors (1 to 50) was used to determine the optimal number of nearest neighbors to use for KNN. According to the scatterplot, n=11 is one of the optimal values to obtain the highest accuracy of KNN (Figure 3). In terms of recall, KNN performance was better for benign predictions than for malignant with a recall score of 0.98 and 0.94 respectively (Table 1).

Support Vector Machine

In terms of F1-score, SVM performed worst for malignant predictions than for benign with an F1-score of 0.94 and 0.97 respectively. In terms of precision, SVM also had a lower performance for malignant predictions than for benign with a precision score of 0.95 and 0.96 respectively. Finally, in terms of recall, SVM performance continue the trend performing worst for malignant predictions than for benign with a recall score of 0.94 and 0.97 respectively (Table 1).

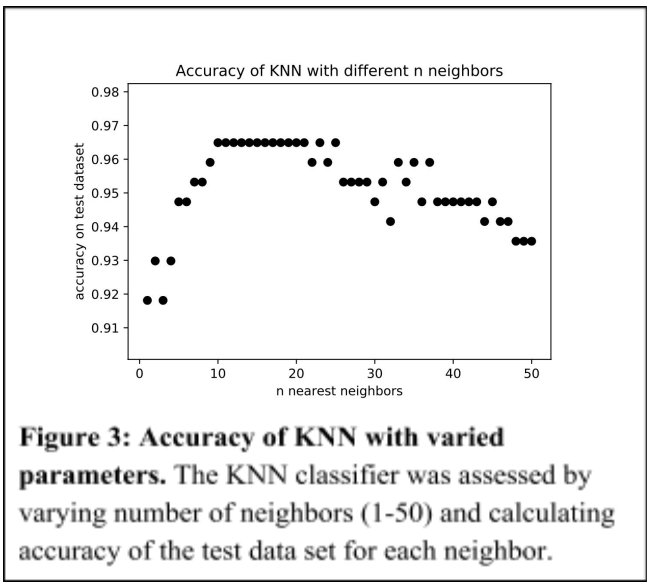


Figure 3: Accuracy of KNN with varied parameters. The KNN classifier was assessed by varying number of neighbors (1-50) and calculating accuracy of the test data set for each neighbor.

Linear Discriminant Analysis

In terms of F1-score, LDA performed worst for malignant predictions than for benign with an F1-score of 0.96 and 0.98 respectively. In terms of precision, LDA had a higher performance for malignant predictions than for benign with a precision score of 1.00 and 0.96 respectively. Finally, in terms of recall, LDA performance was better for bening predictions than for malignant with a recall score of 1.00 and 0.92 respectively. According to the confusion matrix LDA performed well for avoiding to classify false positive samples (Figure 2E).

Conclusion

As part of this project we selected the Wisconsin Diagnostic Breast Cancer Data Set which included 569 tumor samples with 30 different features. Given the importance of diagnosis of malignant tumors, we defined that False Negatives to be worse than False Positives, and consequently, we selected recall as the main performance metric. For our project, we evaluated the performance of five different supervised Machine Learning Classifiers, including: Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbor and Linear Discriminant Analysis.

Among the different classifiers, LR overperformed the others in terms of recall score with a 0.98 on the testing data. In terms of precision, LDA had the highest performance; however, its recall score was only 0.92. Based on our results, we conclude that LR model showed the highest performance based on our selected precision metric.

For future research, we recommend evaluating dimension reduction of the existing features. As it was shown in the exploratory analysis, some features are correlated between each other (eg. area and radius) and consequently, some of them could be dropped from the analysis. For some of the models we also found overfitting (eg. Decision Tree), which suggests the need for dimension reduction and further tuning of the model parameters. In terms of data splitting between testing and training dataset, we recommend to consider the implementation of k-fold cross validation. This splitting procedure allows to have a higher level of confidence of how well the model generalizes to unseen data.

References

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

https://github.com/ctufts/Cheat_Sheets/wiki/Classification-Model-Pros-and-Cons

https://www.datasciencecentral.com/profiles/blogs/logistic-regression-vs-decision-trees-vs-svm-part-ii

https://www.oreilly.com/library/view/machine-learning-with/9781787121515/697c4c5f-1109-4058-8938-d01482389ce3.xhtml

https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, *23*(1), 89-109.

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993, July). Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization* (Vol. 1905, pp. 861-871). International Society for Optics and Photonics.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, *13*, 8-17

Contributions

	Camila	Dianne	Both
Paper	● Methods(1-3, 4a, 4b) ● Results(Decision Tree and Logistic Regression) ● Conclusion	● Abstract ● Introduction ● Methods (4c-4e) ● Results(KNN, SVM, LDA)	
Code	● “Data Modeling” for Decision Tree and Logistic Regression	● “Data Modeling” for KNN, SVM, & LDA	● Data Exploration