A Comparison of Decision Tree and Naïve Based Applied to Breast Cancer Dataset

Khalid Kadri
Department of Computer Science
City, University of London, London, United Kingdom
Khalid.kadri@city.ac.uk

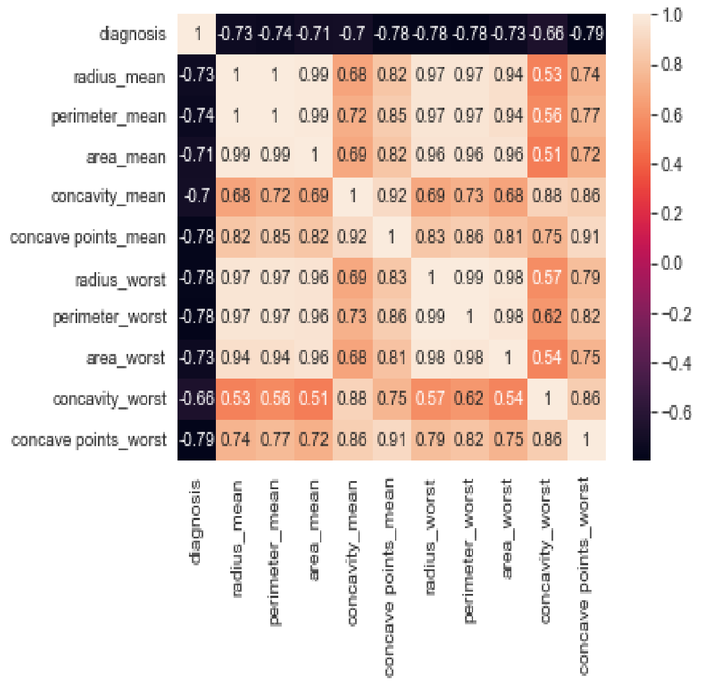
Description and motivation of the problem

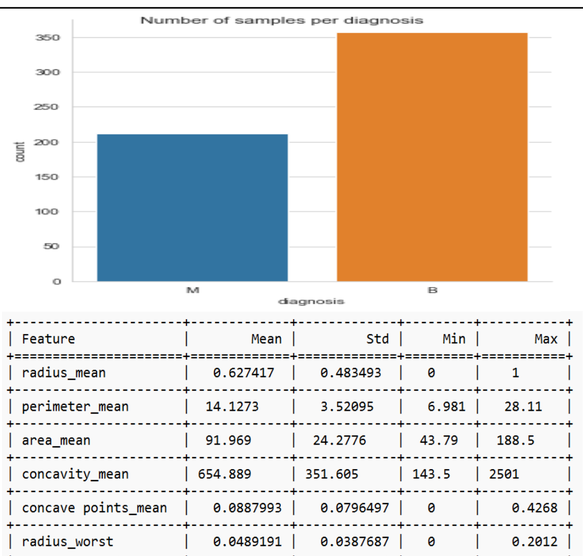
To compare the performance of decision tree and naive Bayes algorithms when applied to a dataset of breast cancer cases. The purpose of this comparison is to evaluate the effectiveness of these algorithms in predicting the presence or absence of breast cancer in patients.

The motivation for this project is to understand which of these algorithms is better suited for breast cancer diagnosis and to identify the strengths and weaknesses of each approach. By comparing the performance of these algorithms on a common dataset, you can gain insight into the characteristics of the data that each algorithm is particularly well-suited to handle, as well as the types of errors each algorithm is prone to make. This information can be used to improve the accuracy and reliability of breast cancer diagnosis, ultimately leading to better patient outcomes.

Initial Analysis Of The Data Set, Including Basic Statistics

- The dataset is a breast cancer sourced from UCI Machine Learning Repository
- The original dataset consists of 570 rows and 32 columns; there are 30 features, one target class and the 'Id', which is irrelevant to our analysis and has no missing values.
- For this analysis, we selected ten features from a dataset of 30 features by choosing the top 10 features that correlated most strongly with the class label (i.e., the 'diagnosis' variable). This is useful because it allows us to identify the most important features for predicting the class label and to potentially reduce the dimensionality of the dataset by removing redundant or highly correlated features.
- For each selected feature in the dataset, we did some basic statistical measures such as the mean, standard deviation, minimum and maximum values to get an idea of the range and distribution of the data
- To visualise the correlations between the features and the class label, we used a heat map, as seen in the diagram.
- To visualise the distribution of the diagnosis classes in the dataset, we created a bar where the plot's x-axis represents the diagnosis classes (i.e., benign or malignant), and the y-axis represents the number of samples in each class. The plot shows that the number of benign samples is slightly higher than the number of malignant samples.





Feature	Mean	Std	Min	Max
radius_mean	0.627417	0.483493	0	1
perimeter_mean	14.1273	3.52095	6.981	28.11
area_mean	91.969	24.2776	43.79	188.5
concavity_mean	654.889	351.605	143.5	2501
concave points_mean	0.0887993	0.0796497	0	0.4268
radius_worst	0.0489191	0.0387687	0	0.2012
perimeter_worst	16.2692	4.82899	7.93	36.04
area_worst	107.261	33.573	50.41	251.2
concavity_worst	880.583	568.856	185.2	4254
concave points_worst	0.272188	0.208441	0	1.252

Table 1. All Feature

Decision Tree (DT)

Decision trees are a type of supervised learning algorithm that can be used for classification and regression tasks. They work by creating a tree-like model of decisions based on feature values.

Pros of using decision trees:

- ✓ They are easy to understand and interpret.
- ✓ The model can be visualised as a tree structure, with each internal node representing a decision based on a feature value and each leaf node representing a prediction. Hence a very resourceful tool for explaining predictions to non-technical stakeholders.
- ✓ Decision trees are relatively fast to train and predict, especially for smaller datasets.
- ✓ They are relatively simple to implement and don't require a lot of pre-processing or feature engineering.
- ✓ Decision trees can handle categorical and numerical data and don't require feature scaling (i.e., normalising the data to a common scale). This makes them a good choice for datasets with different features.

Cons of using decision trees:

- ! It can be prone to overfitting, especially if the tree is deep and complex. Overfitting occurs when the model closely fits the training data, and as a result, it performs poorly on unseen data. To mitigate this, you can prune the tree.
- ! It is not always the most accurate model, especially when the data is noisy or there are complex relationships between the features. (Simplilearn, 2018).
- ! It can be sensitive to small changes in the data, which can result in a different tree being generated each time the model is trained. This can make the model less reliable and less consistent in its predictions.

Naive Bayes (NB)

Naive Bayes is a type of supervised learning algorithm that is based on the Bayes theorem of probability. It is commonly used for classification tasks and can be trained and predicted relatively quickly, even on large datasets.

Pros of using Naive Bayes:

- ✓ It is a relatively simple and fast algorithm that can be trained and predicted relatively quickly, even on large datasets.
- ✓ It is relatively easy to implement and doesn't require a lot of pre-processing or feature engineering.
- ✓ It can handle continuous and discrete data, making it a versatile algorithm for many classification tasks.
- ✓ It can handle missing data in the training data by using the maximum likelihood estimate (MLE) to estimate the probability of each class. This allows the model to continue learning even if some data is missing.

Cons of using Naive Bayes :

- ! It is primarily used for classification tasks and may not perform as well on regression or other types of tasks.
- ! It is a relatively simple algorithm and is not as flexible as other algorithms, such as decision trees or support vector machines. This can make it less effective on complex datasets with many features or intricate relationships between the features.
- ! can be sensitive to uniformly distributed features across the training data. This can cause the model to predict a class more often than is warranted by the data, leading to poor performance on unseen data.

Hypothesis Statement:

- The diagnosis class (i.e., benign or malignant) is significantly correlated with certain features in the dataset, such as the mean radius or the mean concavity.
- The decision tree model will outperform the naive Bayes model on the breast cancer dataset due to its ability to capture non-linear relationships between the features and the diagnosis class.
- The naive Bayes model will outperform the decision tree model on the breast cancer dataset due to its simplicity and ability to handle high-dimensional data.

Methodology:

- Split data into a 90:10 split for train and test data.
- The training set is used to train the model, while the validation set is used to evaluate the model's performance.
- To find the optimal hyperparameters of the models, we used validation accuracies and confusion matrices to select the combination of hyperparameter values that resulted in the most accurate predictions.
- After identifying the best hyperparameters using the validation set, we evaluated the performance of the models on the training set using training, validation accuracies and confusion matrices.
- We compared the performance of the two chosen models by testing them on the testing data using their optimal hyperparameters and contrasted the results to see which model performed better.

Experimental results

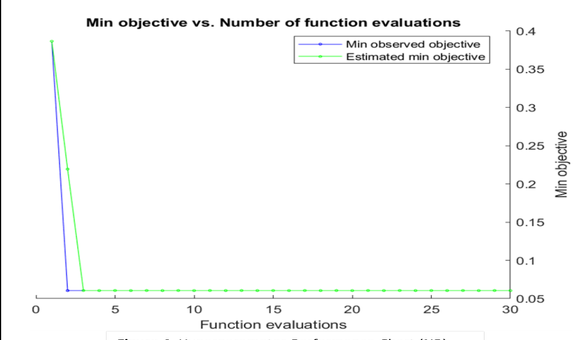


Figure 4. Hyperparameter Performance Chart (NB)

Decision Tree	Model	Naïve Bayes
0.9629	Train set accuracy	0.9395
0.078864	Evaluation time	0.35022
0.9655	Test set accuracy	1
0.9576	Precision	0.9348
0.9640	Recall	0.9373
0.9608	F1 score	0.9360

Figure 3. Classification Results

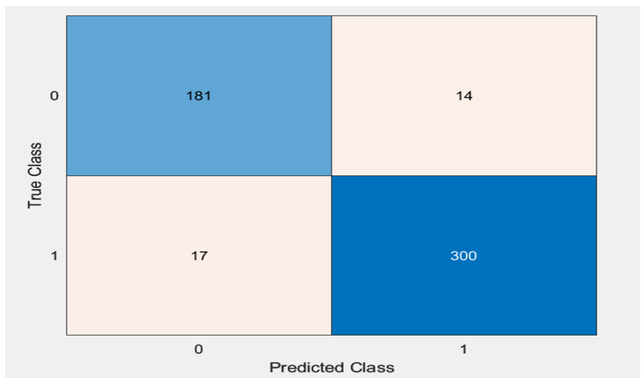


Figure 1. Confusion matrix (NB)

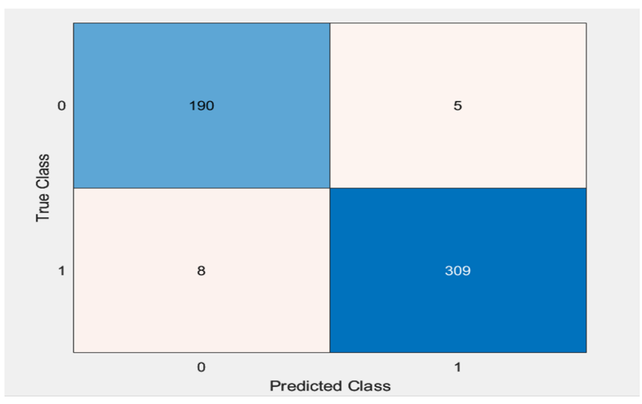


Figure 2. Confusion matrix (DT)

Analysis and Evaluation of results:

- The decision tree model achieved an accuracy of 96.29% on the training set and 96.55% on the test set, with a precision of 95.76%, a recall of 96.40%, and an F1 score of 96.08%. The evaluation time for this model was 0.078864 seconds.
- The naive Bayes model achieved an accuracy of 93.95% on the training set and 100% on the test set, with a precision of 93.48%, a recall of 93.73%, and an F1 score of 93.60%. The evaluation time for this model was 0.35022 seconds.
- Compared to the decision tree model, the naive Bayes model had a lower accuracy on the training set but a higher accuracy on the test set. The naive Bayes model also had slightly lower precision and recall but a slightly higher F1 score.
- The evaluation time for the naive Bayes model was significantly higher than the evaluation time for the decision tree model. However, the higher accuracy on the test set suggests that the naive Bayes model may be a better choice for this dataset.
- Both models achieved relatively high precision, recall, and F1 scores, indicating that they could effectively identify the samples in the positive class while minimising false positives and false negatives.
- The decision tree model had a relatively high accuracy on both the training and test sets, indicating that it could effectively classify the samples in the dataset.
- The naive Bayes model had a slightly lower accuracy on the training set than the decision tree model but a significantly higher accuracy on the test set. This suggests that the naive Bayes model may be more robust and generalisable to unseen data.
- The naive Bayes model had a longer evaluation time than the decision tree model, which may be a consideration when choosing a model for a real-world application with time constraints.

Lessons Learned:

- The naive Bayes model had a higher accuracy on the test set than the decision tree model, despite having a lower accuracy on the training set. This suggests that the naive Bayes model may be more robust and generalisable to unseen data.
- Both models achieved relatively high precision, recall, and F1 scores, indicating that they could effectively identify the samples in the positive class while minimising false positives and false negatives.
- NB features are independent, like a probabilistic model. On the other hand, DT is a non-probabilistic model and just only basic model for classification, which tends to create complex trees leading to poor performance in the real-world testing data

Future Work:

- Investigating the impact of different pre-processing techniques, such as feature scaling or feature selection, on the performance of the models.
- Investigating why DT performs well on the train set but not in the test set, another method like k-fold crossval can be used to check.

References:

altaflab (2020). breast-cancer. [online] GitHub. Available at: https://github.com/altaflab/breast-cancer/blob/master/breast-cancer.pdf [Accessed 22 Dec. 2022].

Wolberg,William, Street,W. & Mangasarian,Olvi. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository.

Simplilearn. (2018). Random Forest Algorithm - Random Forest Explained | Random Forest in Machine Learning | Simplilearn. In www.youtube.com. https://www.youtube.com/watch?v=eM4uJ6XGnSM