Improving the Identification Accuracy of Malignant Tumors: Study on Wisconsin Breast Data Set

You Li[1]

Australian National University

Abstract. This paper implements different approaches for the purpose of predicting malignant breast cancer. A comparison among Artificial Neural Network(ANN), Decision Tree(DT) and Naive Bayes(NB) are applied on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset [3], by measuring their classification test accuracy. The dataset was separated 80% for training phase, and 20% for the testing phase. Results show that all the presented algorithms performed well on the task. In which Decision tree stands out with accuracy 97%.
Further analyze with CNN and evolutionary algorithms on feature selection are investigated. Which feature makes most contribution to the diagnosis.

Keywords: Breast cancer · Tumor · Artificial neural networks · Decision tree · Nave Bayes · WDBC· Deep Learning ·

1 Introduction

Breast cancer is one of the most threatening cancer for women. Abundant of researches were related to such disease. Mammograms have been proved effective in early identification of breast cancer [4],. Utilization of data science in medical fields proves to be assistant in the decision making process of medical practitioners. In this case, studying the dataset and find a way to classify illness cells will be meaningful. The study result can be contribution to further clinical research [1],. The investment task is, given by the features of the cells, make a prediction upon the learning result of classification algorithms. Find the best supervised learning classification model for WDBC dataset.

2 Data Features Analysis

2.1 Figures

Wisconsin Diagnostic Breast Cancer (WDBC) contains 569 instances with 32 attributes, all feature values were recorded with 4 significant digits. The data set was extracted from the computer vision diagnostic system with ten ten different features. Those features were represented with their mean, standard error, worst value [5],. The measurement methods of cells are shown as Figure1 and Figure2 in the appendix.

2.2 Attribute Information of WDBC

1) ID number
2) Diagnosis (M = malignant, B = benign)
3-32): Ten real-valued features are computed for each cell nucleus

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter (The total distance between the snake points constitutes the nuclear perimeter)
d) area (The number of pixels on the interior of the snake adding with 0.5 pixels of perimeter)
e) smoothness (local variation in radius lengths, see appendix)
f) compactness (perimeter^2 / area)
g) concavity (severity of concave portions of the contour, refer appendix)
h) concave points (number of concave portions of the contour)
i) symmetry (The length between longest chord perpendicular to the cell boundary in both directions, see appendix)
j) fractal dimension (perimeter, "coastline approximation", see appendix)
3-12) represents the mean of a-j), 13-22) represents the standard error (SE)of a-j), and the rest represents the worst value of a-j) [5]. More detailed instruction figures were listed in the Appendix [1],.

2.3 Data Processing and Visualization

The Class distribution of WDBC is listed as the Table1.

Table 1. Benign and Malignant classification table.

Class	Count	Percentage
Benign	*357.*	62.7%
Malignant	*212.*	37.3%

Since the attributes are complex, it will be worthwhile to do data visualization and classification of the dataset as the first step. Plot the pair graph for the mean value(Figure1), standard error (see appendix) and the worst value (see appendix) with the respect of 10 features, classify the cells as malignant(blue) and benign(orange). As a result, two cell types are distributed separately. Malignant cells have smaller value than benign cells in each attribute except the fractal dimension. Area, perimeter and radius might be the most significant attributes for classification tasks.

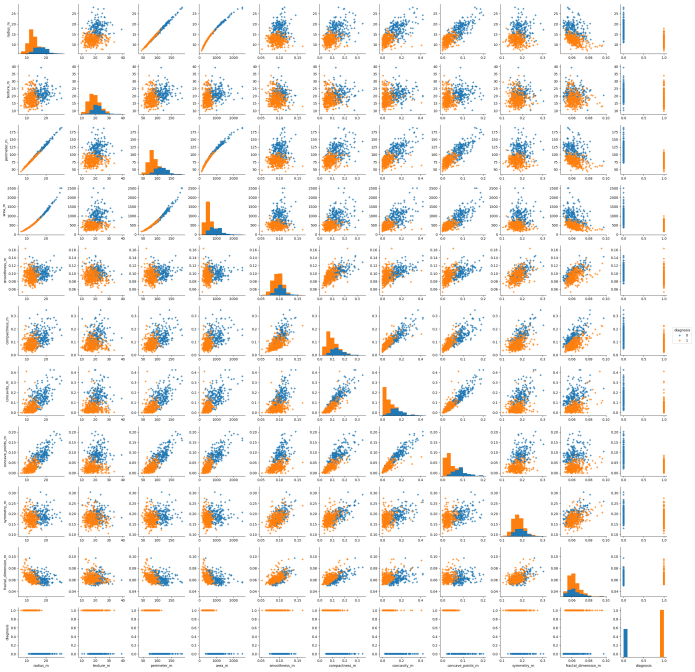

Fig. 1. A figure shows the relationship of attributes for mean values, B for blue and M for orange

3 Learning Models

The data size used for further analysis is (569, 31) without heading. Attribute headers, ids were dropped and the dataset were shuffled. According to original report of the data set, theres no data damage, and all the data are reliable, hence no need for considering the missing data.
Set the M value as 0 and B as 1 for analysis. Separate the data set as train data with 455 instances and test data. Different partitioning size are discussed in further experiment. Target is the diagnosis class, and the rest are imported as features. SGD Artificial Network, Decision Trees, Naive Bayes algorithms were implemented for this supervised classification task. In addition, convolution neural network and feature selection methods also approached for classification task.

3.1 Artificial Neural Network with SGD

Build the neural network with Multi-layer Perceptron classifier is designed to be used as a learning method. This model optimizes the log-loss function using LBFGS or stochastic gradient descent. [3] Structure of this neural network shown as Figure. The network will be trained with Stochastic Gradient Descent (SGD) as an optimizer, holding the current state and will update the parameters based on the computed gradients and minimizes the loss function during network training. The train result will be evaluated by cross entropy. The performance of this neural network is evaluated by the accuracy of its prediction of the cells [6].

3.2 Decision Trees

A binary decision tree could be a suitable model as a classifier for separate targets according to the value of each features. M type and B type cells are distributed separately of each features in the data set. Initialize the decision tree with entropy criterion as this trained for information gain [3]. Train the data until the leaf is pure. The best prediction accuracy through experiment is 95.37

3.3 Naive Bayes

Consider the Naive Bayes learning algorithm with the assumption that every pair of features are independent. This is the method that based on probability theory, which reduce the noise of the data. This algorithm make prediction by calculate conditional probability of each features [2, 3], and make decision based on the calculated probability. This algorithm have good performance on used for multi class prediction.

$$P(c_i|A) = \frac{P(A|c_i)p(c_i)}{P(A)} \tag{1}$$

$$P(A|c_i)p(c_i) = \prod_{j=1}^{n} P(a_j|c_i)P(c_i) \tag{2}$$

Implement this learning algorithm is similar as the previous two approaches. Its obvious that Naive Bayes runs much faster than previous training algorithm. The training result is unstable but floats around 90% accuracy.

3.4 Convolution Neural Network

This deep learning method is implemented to gain higher performance on prediction. Instead of RNN and LTSM, CNN is more suitable for this classification task. This network is built based on the neural network, expanded with two convolution layers, one pooling layer and one dense layer in the hidden layers. Rectify functions were applied for convolution layer, restrict the value between -1 to 1, and pick the best value. For the output layer, softmax function is applied to converts vectors into class probabilities.
During the training, accuracy increases with the epoch flow up from 0 to 2000, moves steady around 90% when 100 epochs reached.

3.5 Feature Selection

Feature selection has advantage on limit the number of input features in a classifier in order to have both good prediction and less computationally intensive model. F score is used to calculate the importance of each feature. An reliable feature selection method using grid search was bring forwarded. F score of each attributes were calculated and generate subsets with feature at highest F scores. [8].The selected attributes will be experiment second time as an validation of feature pruning. In this case, the selected 2)-12) attributes could reach among 70% accuracy.

$$F(i) = \frac{(\overline{x}_i{}^{(+)} - \overline{x}_i)^2 + (\overline{x}_i{}^{(-)} - \overline{x}_i)^2}{\frac{1}{n_+ - 1}} \sum_{k=1}^{n_+} (x_{k,i}^{(+)} - \overline{x}_i^{(+)})^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} (x_{k,j}^{(-)} - \overline{x}_i^{(-)})^2 \tag{3}$$

4 Evaluation method

Confusion matrix and 10 fold validation are applied for result evaluation, As well as the ROC curve.. To specify, classification result confusion matrix is defined as following [8]. Precision, recall. accuracy and F_1 score are calculated depend on those values. In this paper, classification result with B is positive, M means negative.

Table 2. Confusion matrix representation

Fact	Predicted Positive	Negative
Positive	True positive	False negative
Negative	False positive	True negative

$$Precision = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{4}$$

$$Recall = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{5}$$

$$Accuracy = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}} \tag{6}$$

$$F_1 = \frac{2 \times \mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}} \tag{7}$$

5 Results and Discussion

5.1 Artificial Neural Network

The iteration size, learning rate and hidden layer size will be significant influence the learning outcome. Table2 indicates sample results with respect of different values on 80% training set and 20% testing set . Hence we can drive a conclusion that the learning accuracy in direct proportion to the complexity of the neural network.

Table 3. Learning performance of ANN

Active Function	entropy	learning rate	hidden layer size	Accuracy
identity	*100.*	1	30	60.526%
identity	*20000.*	1e-5	(100,100)	92.982%
relu	*100.*	1	30	64.035%
relu	*20000.*	1e-5	(100,100)	92.982%
logistic	*100.*	1	30	85.088%
logistic	*20000.*	1e-5	(100,100)	91.228%
tanh	*100.*	1	30	87.719%
tanh	*20000.*	1e-5	(100,100)	92.105%

5.2 Dataset Partition

Experiments on dataset partition were did in previous studies. Similarly, in this paper, the train data and test data will be separated with proportion on 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. The experiments are as follow. This result shows that the data partition is better trained with more valid training data. 90% training data will have best performance while 80% data would be most representative. Too less test data might result in accidental mistake, while in less training data may lead to insufficient learning of the neural network [7].

Table 4. partition size

	Accuracy								
partition of training data	10%	20%	30%	40%	50%	60%	70%	80%	90%
ANN	91%	87%	92%	94%	92%	95%	94%	92%	96%
Decision Tree	90%	91%	93%	91%	92%	93%	95%	95%	96%
Naive Bayes	89%	91%	91%	88%	89%	87%	89%	88%	86%
CNN	84%	84%	86%	85%	88%	86%	90%	92%	94%

5.3 Classifiers Comparisons

Following table summarized the performance of four training method with 80% training data. Decision tree could be the most appropriate training algorithm with both high performance on accuracy, precision, recall and low time consumption. The terminal record could be checked at the appendix. So far, this paper

Table 5. Training result evaluation

Method	Precision	Recall	F_1 score	Accuracy
ANN	90.6%	97.1%	93.7%	92.1%
DT	95.6%	91.5%	93.5%	92.1%
NB	85.5%	98.6%	91.8%	88.5%
CNN	93.3%	85.7%	89.3%	91.2%

discussed implementation and test learning models of neural networks and CNN. The experiments result shown similar performance of published papers around 90% accuracy [2, 6]. The task of implementation and comparison could be regarded as a completion. This paper mainly uses the partition of 0.8 is due to this data separation has the best representation of the whole dataset.

6 Conclusion and Future Work

This work suggests that the constitutional neural network could be the most suitable learning algorithm for classify breast cancer according to the diagnosed data. Since the experimental results are generated separately, there exists coincidence. Future work will explore the potential to eliminate such disambiguation [6]. The paper focus on supervised learning algorithm, hence unsupervised algorithm such as Nearest Neighbor could be considered as future work. Additionally, the completed implementation of the neural network could also be addressed in the future.

References

1. Abien Fred Agarap, On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset, 2017
2. Chotirat Ann and Dimitrios Gunopulos, Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection, 2002.
3. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
4. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian, Breast Cancer Wisconsin (Diagnostic) Data Set, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29, 1995
5. W. Nick Street, W. H. Wolberg, O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis", Proc. SPIE 1905, Biomedical Image Processing and Biomedical Visualization, (29 July 1993); https://doi.org/10.1117/12.148698 https://doi.org/10.1117/12.148698
6. W. Nick Street, A Neural Network Model for Prognostic Prediction, 1998
7. Shajib Ghosh, Jubaer Hossain, Shaikh Anowarul Fattah, Celia Shahnaz, Asir Intisar Khan, "Efficient Approaches for Accuracy Improvement of Breast Cancer Classification Using Wisconsin Database", 2017
8. Mehmet Fatih Akay, "Support vector machines combined with feature selection for breast cancer diagnosis", 2009

Appendix

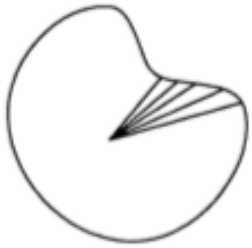

Fig. 2. radial lines used for smoothness

Fig. 3. chords used to compute concavity

Fig. 4. segments used in symmetry

Fig. 5. factal dimension

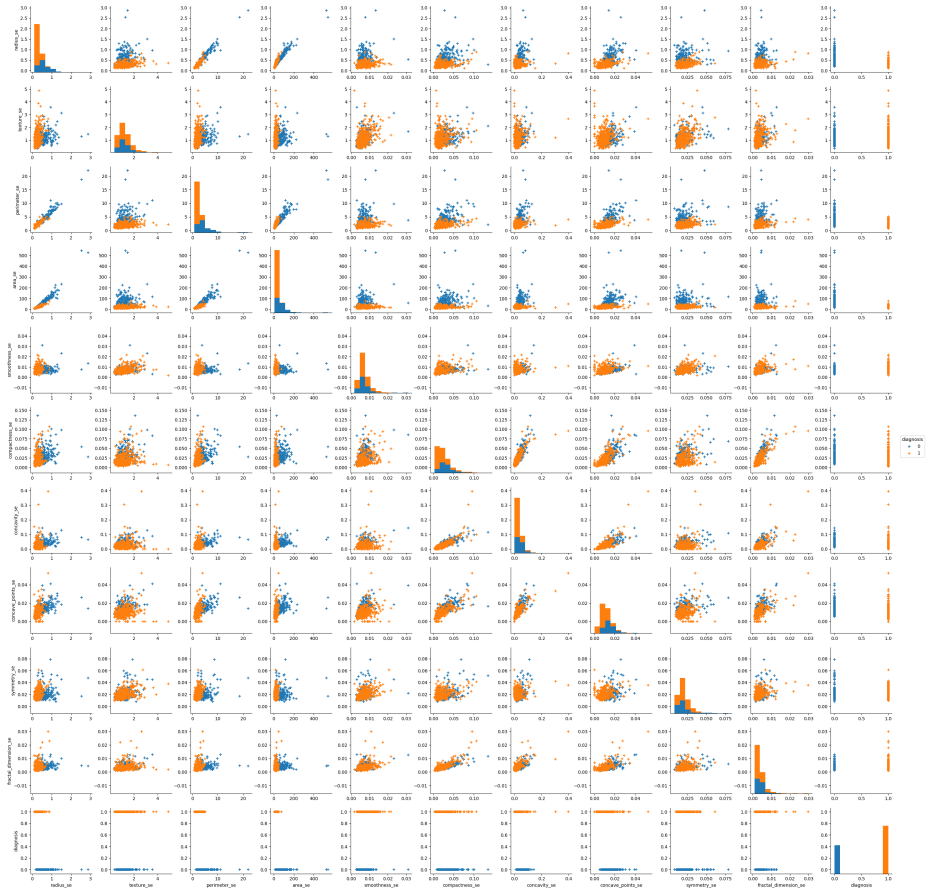

Fig. 6. A figure shows the relationship of attributes for standard error values, B for blue and M for orange

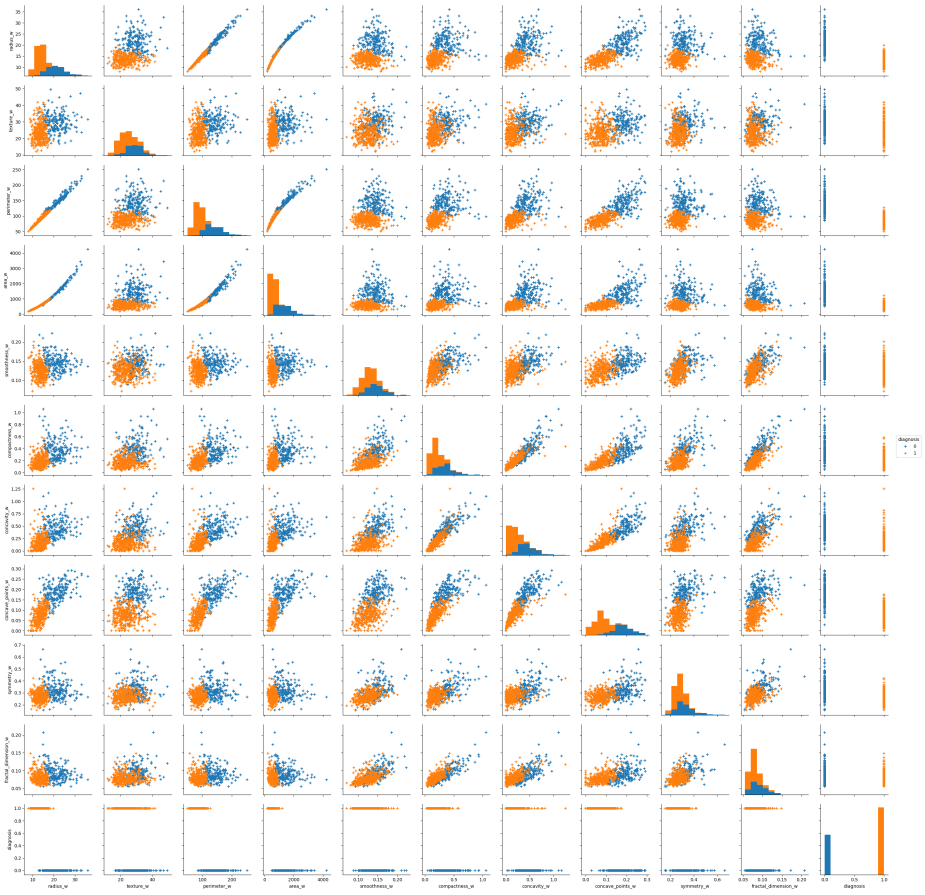

Fig. 7. A figure shows the relationship of attributes for worse values, B for blue and M for orange

```
Ceasar% python3 task01_ann.py
train data: 455 test data: 114
true positive: 68
true negative: 37
false positive: 7
false negative: 2
precision: 0.9066666666666666
recall: 0.9714285714285714
f1_score: 0.9379310344827586
accuracy: 0.9210526315789473
Ceasar% python3 task02_decition_tree.py
train data: 455 test data: 114
true positive: 65
true negative: 40
false positive: 3
false negative: 6
precision: 0.9558823529411765
recall: 0.9154929577464789
f1_score: 0.9352517985611511
accuracy: 0.9210526315789473
Ceasar% python3 task03_naive_bayes.py
train data: 455 test data: 114
true positive: 73
true negative: 28
false positive: 12
false negative: 1
precision: 0.8588235294117647
recall: 0.9864864864864865
f1_score: 0.9182389937106918
accuracy: 0.8859649122807017
Ceasar% python3 task04_cnn.py
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/sk
learn/cross_validation.py:41: DeprecationWarning: This module was deprecated in
version 0.18 in favor of the model_selection module into which all the refactore
d classes and functions are moved. Also note that the interface of the new CV it
erators are different from that of this module. This module will be removed in 0
.20.
  "This module will be removed in 0.20.", DeprecationWarning)
train data: 455 test data: 114
true positive: 42
true negative: 62
false positive: 3
false negative: 7
precision: 0.9333333333333333
recall: 0.8571428571428571
f1_score: 0.8936170212765957
accuracy: 0.9122807017543859
```

Fig. 8. terminal record