Breast cancer diagnosis

Meysam Davoudi
Okan university (computer department)
Istanbul, Turkey
medavoudi@stu.okan.edu.tr

Abstract **In this review, we use the breast cancer dataset [1] available in the scikit-learn library to train 7 machine learning models. We then calculate the accuracy score, precision score, recall score, and confusion matrix for these models using this dataset. We compare them and finally suggest the optimal model. The models used in this review are Naive Bayes, KNN, Decision Tree, Random Forest, SVM, Logistic Regression, and ANN**

Keywords breast, cancer

I. Introduction

With the rapid advancement in information technology and the enormous increase in data volume, research in the field of medical diagnosis and prediction of diseases has become an essential and challenging need. At this juncture, breast cancer, as one of the most common and complex diseases in women, has gained significant importance, and the development of optimal methods for its early and effective diagnosis seems essential.

The main objective of this research is to use machine learning techniques in analyzing data related to breast cancer and improving the accuracy of diagnosis. Given the complexity of the diagnosis process of this disease, we, as researchers, aim to improve the accuracy, validity, and reliability of machine learning models. This research promotes the combination of technology and medicine to provide innovative approaches in medical data analysis.

this research will provide a detailed explanation of the proposed methods, the models used, and the evaluation of the results.

Finally, we hope that the results of this research will not only help improve the accuracy of breast cancer diagnosis but also contribute significantly to achieving a remarkable progress in the field of medical data analysis and prevention of this disease.

II. Methodology

A. *Dataset*

In this research, we used the breast cancer dataset available in the scikit-learn library. This dataset has two classes: malignant and benign, which are represented by the numbers 0 and 1, respectively. The dataset has 30 features. The features available in this dataset are as follows:

1. mean radius: the average radius of the cell nuclei.

2. mean texture: the average texture of the cell nuclei in the images of the tissue.

3. mean perimeter: the average perimeter of the cell nuclei.

4. mean area: the average area of the cell nuclei.

5. mean smoothness: the average smoothness or roughness of the cell nuclei.

6. mean compactness: the average compactness of the cell nuclei.

7. mean concavity: the average concavity of the cell nuclei.

8. mean concave points: the average number of concave points in the cell nuclei.

9. mean symmetry: the average symmetry of the cell nuclei.

10. mean fractal dimension: the average fractal dimension of the cell nuclei.

For each of these features, there are also “standard error” and “worst” versions, which represent the standard error and the worst value of the feature, respectively.

B. *Preprocessing*

First, we divided the research data into two parts of training and testing with a ratio of 80:20. This was done using the following code:

```
from sklearn.model_selection import
train_test_split

train_feature, test_feature, train_label,
test_label = train_test_split(bc.data, bc.target,
test_size=0.2, random_state = 42)
```

Then, in order to optimally use machine learning models, we normalized the training and test data to reduce the effect of different scales of variables on the performance of the model. This was done using MinMaxScaler:

```
from sklearn.preprocessing import MinMaxScaler

scalar = MinMaxScaler(feature_range=(0, 1))

train_feature =
scalar.fit_transform(train_feature)

test_feature = scalar.transform(test_feature)
```

These pre-processing steps prepare data for applying machine learning models with higher accuracy and efficiency.

C. *Classification*

To improve the accuracy and efficiency of breast cancer diagnosis, various machine learning models have been used. The aim of this research is to select appropriate models based on the nature of the data and optimize their parameters to achieve more accurate and interpretable results.

The following list shows the different machine learning models used:

1. Gaussian Naive Bayes (GNB):

Uses Gaussian distribution for quick and efficient decision-making. No special parameters need to be adjusted.

```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(train_feature, train_label)
```

2. K-Neighbors Classifier (KNN):

Applies the k-nearest neighbors algorithm with optimized number of neighbors and uses the kd_tree structure. Parameters: n_neighbors=8, algorithm='kd_tree', leaf_size=28.

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=8,
algorithm='kd_tree', leaf_size=28)
knn.fit(train_feature, train_label)
```

3. Decision Tree Classifier (DT):

Uses decision tree with control over depth and number of samples to prevent overfitting. Parameters: max_depth=64, min_samples_split=2, criterion='gini', random_state=42.

```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(max_depth=64,
min_samples_split=2, criterion='gini',
random_state=42)
dt.fit(train_feature, train_label)
```

4. Random Forest Classifier (RF):

Uses decision forest to combine decisions of multiple trees to increase accuracy. Parameters: n_estimators=1000, max_depth=32, min_samples_split=4, random_state=42.

```
from sklearn.ensemble import
RandomForestClassifier

rf = RandomForestClassifier(n_estimators=1000,
max_depth=32, min_samples_split=4,
random_state=42)
rf.fit(train_feature, train_label)
```

5. Support Vector Machine (SVM):

Applies SVM model using a polynomial kernel to model complex data. Parameters: kernel='poly', random_state=42.

```
from  sklearn.svm import SVC

svm = SVC(kernel='poly', random_state=42)
svm.fit(train_feature, train_label)
```

6. Logistic Regression (LR):

Uses logistic regression method to analyze the effect of variables on the probability of prediction.

Parameters: random_state=42.

```
from sklearn.linear_model import
LogisticRegression

lr = LogisticRegression(random_state=42)
lr.fit(train_feature, train_label)
```

7. Artificial Neural Network (ANN):

Executes a neural network with hidden layers to increase learning capabilities. Parameters: hidden_layer_sizes=256, activation='relu', solver='adam', batch_size=64, max_iter=1000, random_state=42.

```
from sklearn.neural_network import MLPClassifier

ann = MLPClassifier(hidden_layer_sizes=256,
activation='relu', solver='adam', batch_size=64,
max_iter=1000, random_state=42)

ann.fit(train_feature, train_label)
```

III. Results

The results of executing various machine learning models on breast cancer data are presented in tables I and II. The main evaluation metrics, including accuracy, precision, recall, and confusion matrix, are provided for each model.

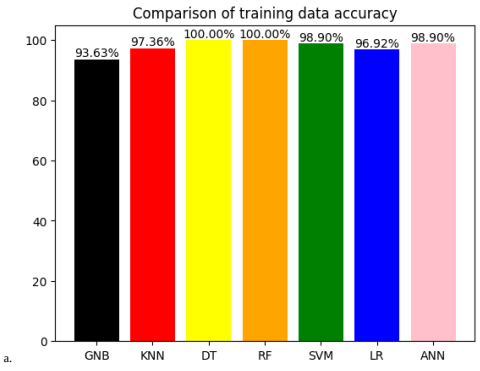
TABLE I.

Model	**evaluation metrics**			
	Accuracy		***Precision***	***Recall***
	train	***test***		
GNB	0.936263	0.964912	0.958904	0.985915
KNN	0.973626	0.964912	0.971830	0.971830
DT	1.0	0.947368	0.957746	0.957746
RF	1.0	0.964912	0.958904	0.985915
SVM	0.989010	0.982456	0.972602	1.0
LR	0.969230	0.982456	0.972602	1.0
ANN	0.989010	0.973684	0.972222	0.985915

TABLE II.

Model	**confusion matrix**			
	TP	***TN***	***FP***	***FN***
GNB	40	70	3	1
KNN	41	69	2	2
DT	40	68	3	3
RF	40	70	3	1
SVM	41	71	2	0
LR	41	71	2	0
ANN	41	70	2	1

The bar charts below vividly depict the performance of each model across each criterion.



a.

Figure 1: Comparison of training data accuracy

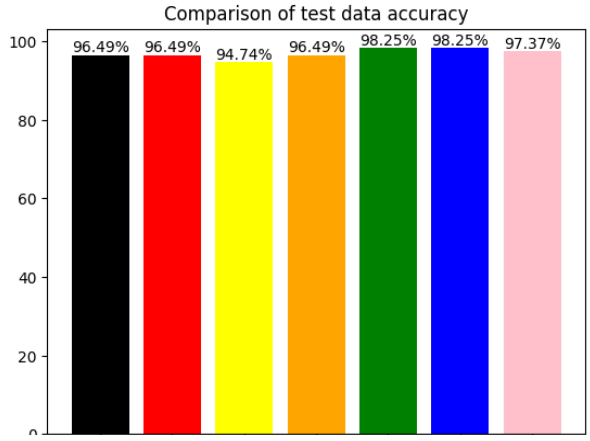


Figure 2: Comparison of test data accuracy

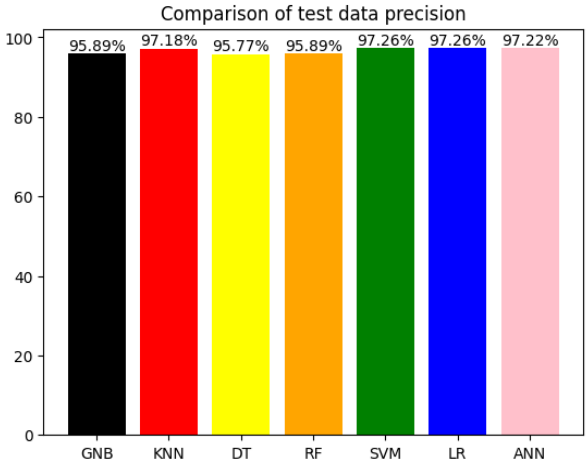


Figure 3: Comparison of test data precision

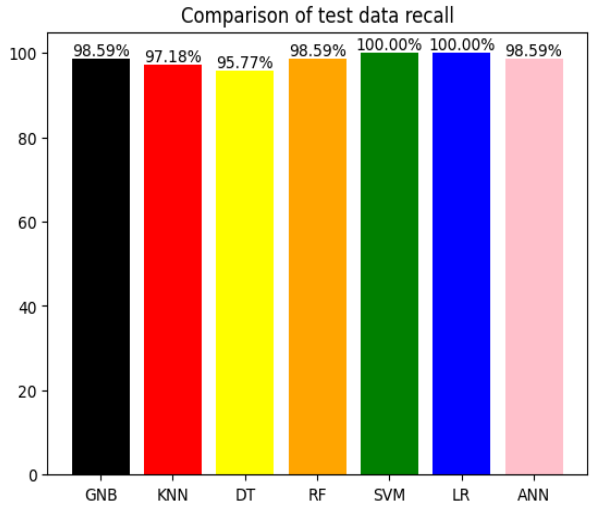


Figure 4: Comparison of test data recall

IV. Discussion

As you can see in Table I, GNB, KNN, DT, RF, SVM, LR and ANN models are trained with different accuracies. To compare the performance of these models, various evaluation criteria such as accuracy, precision, recall and confusion matrix have been calculated for each model.

According to Table I , SVM and KNN models have the lowest difference in accuracy between training and test data.

(SVM) 0.989010 [train] - 0.982456 [test] = 0.006554

(KNN) 0.973626 [train] - 0.964912 [test] = 0.008714

And also, according to Table II, SVM and LR models with a false negative value equal to zero showed the best performance.

Conclusion

According to the analysis of the evaluation results, the SVM model has shown the best performance for breast cancer diagnosis with the lowest accuracy difference between the training and test data, as well as the false negative value equal to zero.

References

[1] UCI ML Breast Cancer Wisconsin (Diagnostic) datasets

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870,San Jose, CA, 1993.

- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.

- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.