Breast cancer prediction modeling

Discovery of biomarker signatures with machine learning

Mid Eum Lee, PhD

April, 2020

Table of contents

1. Introduction

1.1 Background

Breast cancer is the most common cancer type in women. Early diagnosis with accuracy and precision is crucial for improving survival. According to Danish Cancer Society[1], 61% respondents reported that 24% errors in cancer diagnosis are happening due to doctors' assessment errors.

In this study, we will use machine learning algorithms to improve the method of breast cancer prediction. The results obtained in this analysis will be helpful to doctors as well as laboratory experts and researchers for further improvement in their decision-making. The accurate prediction at an early stage of tumor progression will be critical for saving patients lives.

1.2 Data

The dataset was released by UCI machine learning repository. This dataset includes features of cells that are obtained from a fine needle aspirate(FNA) of a breast mass. These cells were analyzed in digital images after cell nuclei were stained for visualization. Some of the original images can be found in the original database[2]. Attribute Information in the dataset was the ID number and Diagnosis (M = malignant, B = benign). 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
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

---

[1] "Cancer patients' experiences of error and consequences ...." https://pxjournal.org/cgi/viewcontent.cgi?article=1039&context=journal.
[2] "Breast Cancer Wisconsin ... - UCI Machine Learning Repository." 1 Nov. 1995, http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29.

1.3 Workflow of modeling overview

To find the best method for prediction, machine learning modeling is applied in the dataset. Figure 1 describes the overview of the machine learning process. The datasets collected from various resources will be used for modeling. There will be some missing data, unorganized data and unnecessary data. The data preprocessing is an important step to build the modeling. In this step, we clean the data and convert it into a clear dataset. To perform the pre-processing of the raw data for machine learning, we can take the following steps:

1) Convert data type to numeric features because categorical and ordinal data cannot be handled by machine learning (ML) model.
2) Remove or ignore the missing data.
3) Fill up the missing values by manual typing with mean or highest frequency values.
4) Predict the missing data using machine learning model
5) Find the outliers that deviate strongly from the other values in the dataset.

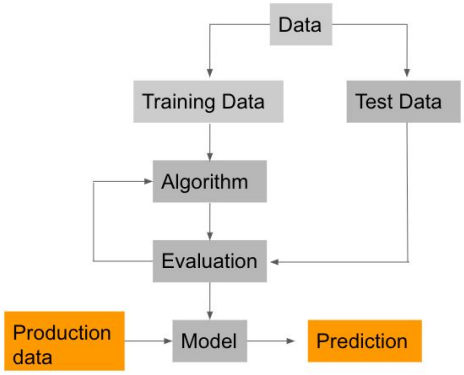

Figure 1. overview of the workflow of ML

Once the dataset is ready, the model will be selected. There are two categories: one is supervised learning and the another is unsupervised learning. In supervised learning, there are two categories. One is classification and the another is regression. Classification is the question of classifying groups with categorical variables. For example, the output of cancer prediction will be either “Benign” or “Malignant”.While a regression is the problem when the variable is continuous (i.e. the output is numeric variables). In unsupervised learning, “Clustering” is the example. For clustering, datasets are divided into groups but data is not labeled unlike classification.

For training the model, we split the dataset into “training data” and “testing data”. Machine learning algorithms will be used to train dataset only using “training data” and the unseen data will be used to assess the performance on the test set.

2. Data Exploration

2.1. Data cleaning

There are 569 entries (rows) for each column and total 33 columns are included in the original dataset. One column contains all null values with 569 entries. This unnamed column was removed as well as the ID column. So the total 31 columns with 569 entries were used for the analysis. The details regarding the selection of 31 columns is discussed in the IPython notebook.

All features in this dataset will be explored to see their relationship with cancer prediction class. Details regarding each feature can be found in UCI website[3]. The target column will be the diagnosis column, which contains two values: 1 for malignant and 0 for benign.

2.2. Breast cancer prediction

Breast cancer is the most frequent cancer in women worldwide, with 1.67 million new cases diagnosed in 2012[4]. As diagnosis error is the largest category of medical claims in the United States, 75% of all breast cancer patients files malpractice suits under age 45 (median, 42)[5].

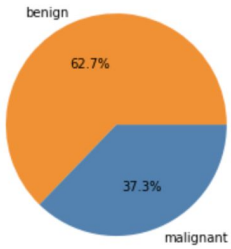

Figure 2. Diagnosis classification

There are several diagnostic methods for breast cancer diagnosis including mammogram and biopsies. Particularly with biopsies, there's a risk for infection and other complications. As this dataset is from the images of biopsy samples, the analysis needs to be accurate for the proper treatment and to get meaningful diagnostic results from biopsies.

This dataset contains 357 Benign and 212 malignant data. In Figure 1, There is 62.7% classification category in benign cases. Finding critical features for diagnosis will be needed for better classification in this dataset.

---

[3] "Breast Cancer Wisconsin (Diagnostic) - UCI Machine Learning." 1 Nov. 1995, http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.
[4] "Cancer incidence and mortality worldwide: sources, methods ...." 9 Oct. 2014, https://www.ncbi.nlm.nih.gov/pubmed/25220842.
[5] "Breast Cancer Misdiagnosis Most Frequent Cause of ...." 1 Sep. 1999, https://www.cancernetwork.com/breast-cancer/breast-cancer-misdiagnosis-most-frequent-cause-malpractice-suits.

2.3. Feature analysis

For further analysis, selected features will be considered for breast cancer classification problems. In figure 3, we see the overall patterns of feature distribution categorized by diagnosis.

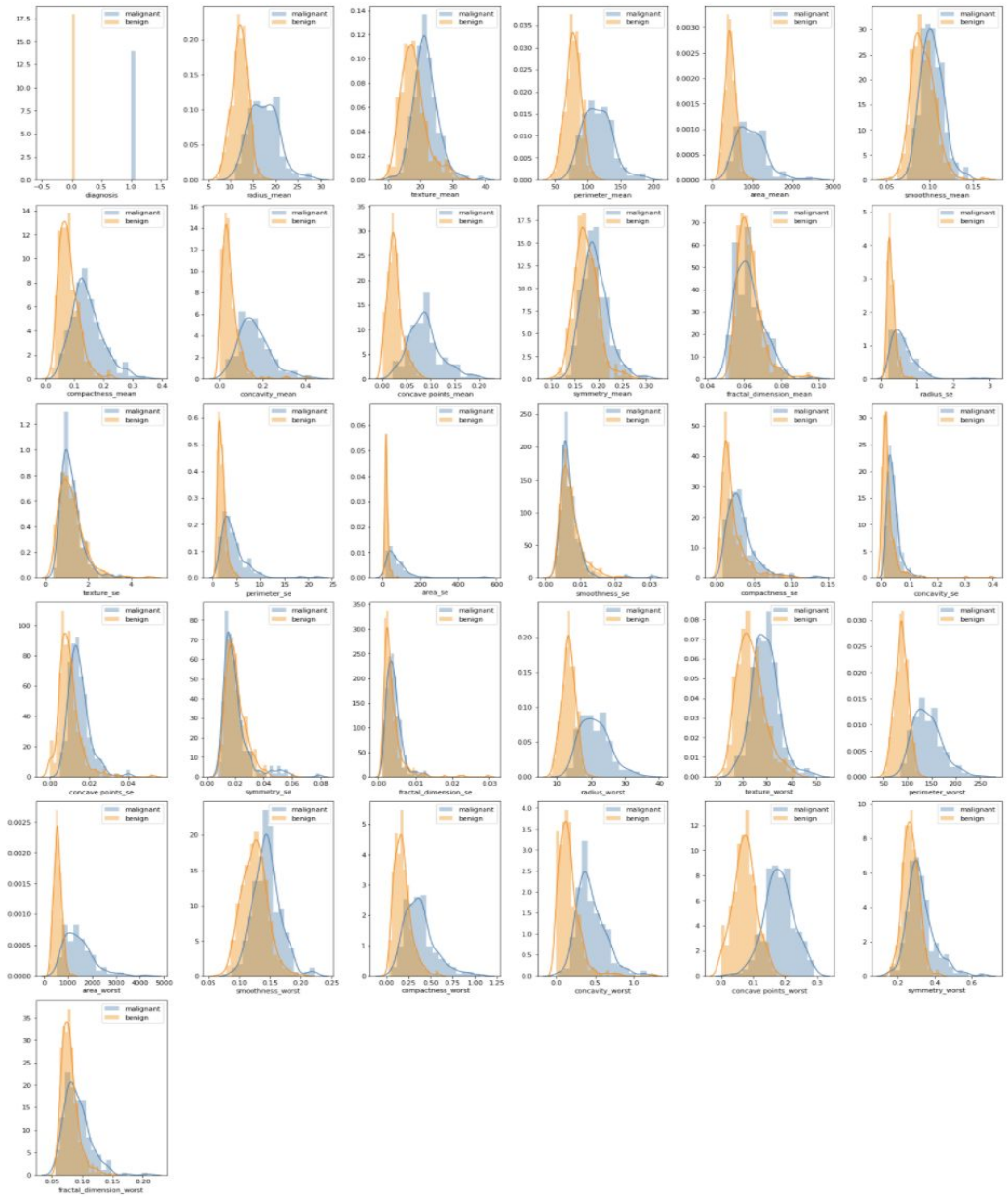

Figure 3. Distribution of features categorized by prediction classes.

In figure 3, most features can be differentiated by prediction classes. We can see that “fractal_dimension_mean”, “texture_se”, “smoothness_se”, “symmetry_se”, “fractal_dimension_se” can't differentiate malignant versus benign.

In order to examine the spread of variables more carefully, boxplots were used to check skewness in the data. Box plots clearly show the outliers. As an example, “radius_mean” and “texture_mean” were plotted and categorized by diagnosis (see figure 4). Student t-test shows that “texture_mean” values in the benign group vs. malignant group are statistically different. But the box plots show that there are many outliers in “texture_mean” variables compared to “radius mean”. Box plots show that the “radius_mean” variable is much better compared to “texture_mean” for prediction.

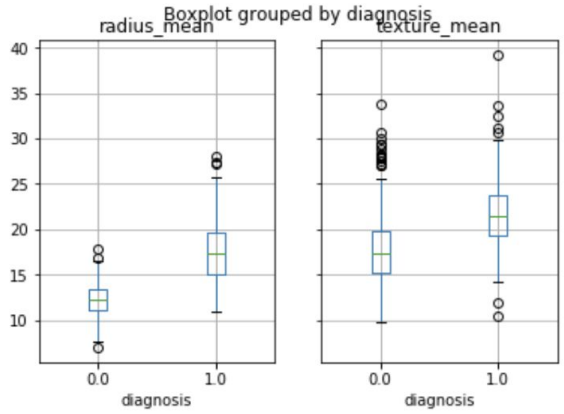

Figure 4. Box plots of radius_mean (left) and texture_mean(right) categorized by diagnosis.

Additionally, there is not much difference in “smoothness of mass” between groups (Figure 5a). The “compactness_mean” is higher in the malignancy group. We also see that there are many outliers in concavity_mean (Figure5b). Concave points_mean is higher in the malignancy group(Figure5b).

a.

b.

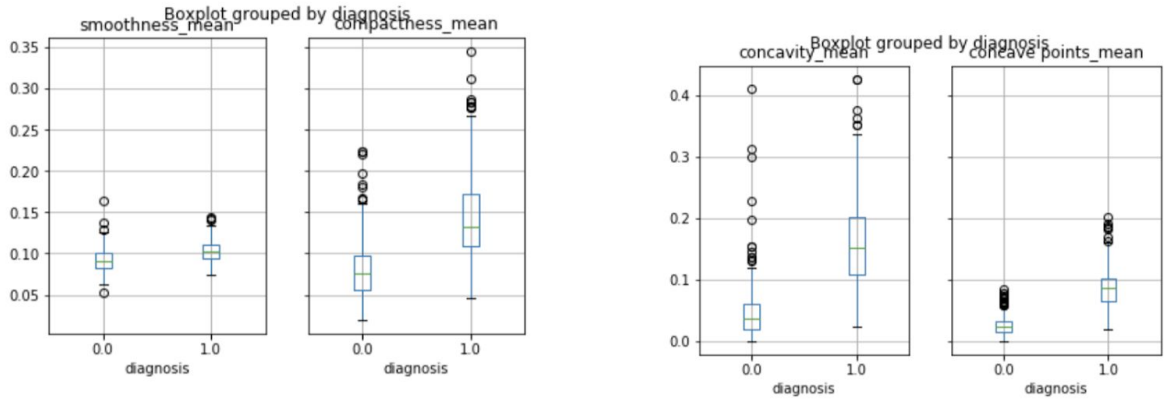

Figure 5. Box plots of features categorized by diagnosis.

In figure 6, the perimeter and the area of mass are higher in the malignancy group. These box plots are great to see the outliers and get features that are more reasonable for prediction.

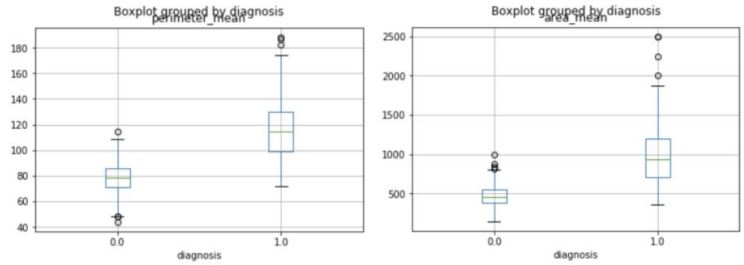

Figure 6. Box plots of perimeter_mean and area_mean categorized by diagnosis.

2.4. Correlation analysis

Multiple features in the dataset are analyzed for the correlation. Correlation matrix plot is a good way to visualize the correlations among features in columns.

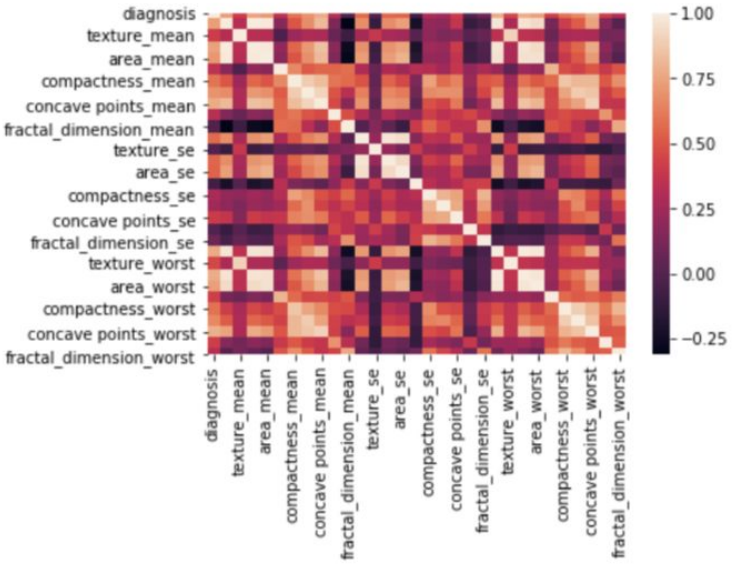

Figure 7. Correlation heatmap of datasets.

Based on the correlation map, the highly correlated features were plotted in Figure 8.

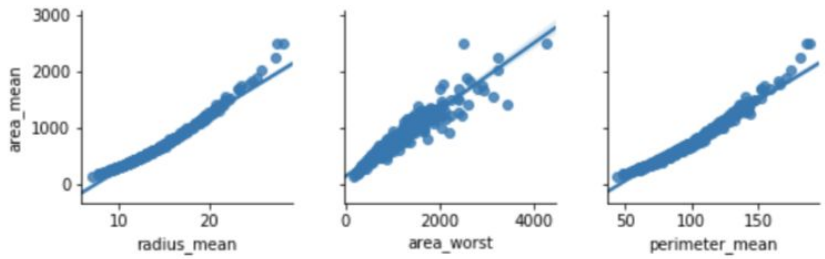

Figure 8. Linear correlation of area_mean and radius_mean, area_worst, perimeter_mean.

The regression line shows clear linear correlations between area_mean and radius_mean, area_worst, perimeter_mean features. The features that are not highly correlated are shown in figure 8.

In contrast to Figure 8, data points are highly scattered and the slope (tangent of angle) of the regression lines are smaller than the slope in figure 9. The linear line slope is going flat in 'fractal_dimension_worst' feature. Area_mean feature is not highly correlated with 'texture_worst','texture_mean','compactness_worst','concave points_worst' and 'fractal_dimension_worst'.

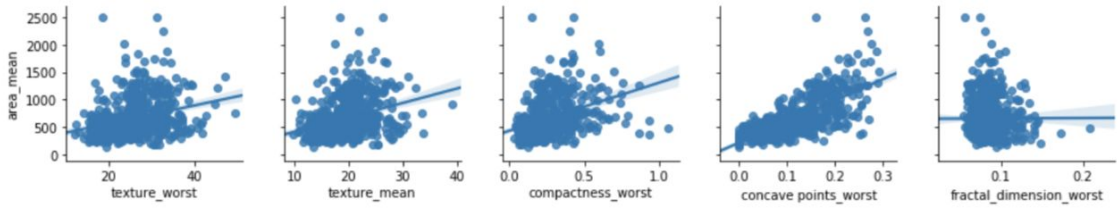

Figure 9. Correlation of area_mean and texture_worst, texture_mean, compactness_worst, concave points_worst, fractal_dimension_worst.

3. Modeling

As the data in the diagnosis column is labeled with 1 for malignancy and 0 for benign, this dataset will be handled as a classification problem. Supervised machine learning algorithms are used to predict the model.

3.1. Data Pre-processing

Before running machine learning algorithms, there are some data pre-processing steps that are needed for further analysis. First, label encoding is performed to change all categorical variables into numeric variables. For logistic regression and support vector machine algorithms(SVM), we need to scale our data. When we collect data and perform feature extractions, each feature is presented in a different scale. In order to run these modeling, we should perform feature scaling and mean normalization for reducing the impact of large valued features and allowing small valued features to contribute equally.

In this report, Standardscaler in sklearn is used for scaling. StandardScaler is based on z score, and it is calculated by " z = x-u(mean of training set) /s (standard deviation of training set)". Next, data was splitted into two groups: training data (90%), testing data (10%). Scaled data was used in logistic regression and support vector machine modeling.

3.2. Logistic Regression

Logistic regression is a model that utilizes a logistic function to test a binary variable. To evaluate the performance, confusion matrix and classification report were performed. Confusion matrix describes the performance of a classification model on test data. It is a table of combinations with predicted and actual values. In the confusion matrix(see figure 10), True Positive (TP) indicates positive and it is true. True Negative (TN) is predicted 'negative' and it's true. False Positive (FP) is the predicted positive but it's false. False Negative (FN) is the predicted negative but it's false.

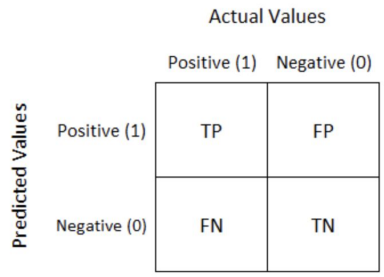

Figure 10. Confusion matrix in machine learning

confusion matrix:

35	0
1	21

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.97	1.00	0.99	35
1	1.00	0.95	0.98	22
accuracy			0.98	57
macro avg	0.99	0.98	0.98	57
weighted avg	0.98	0.98	0.98	57

Figure 11. Confusion matrix (left) and classification report (right) from logistic regression model.

In classification reports in sklearn, precision indicates accuracy of positive production. Precision is defined as the ratio of true positive to the sum of true positive and false positive. Recall is the true positive that is correctly identified. Recall is defined as the ratio of true positive to the sum of true positive and false negative. F1-score is calculated with precision and recall. F1 score calculation formula is 2*(Recall * Precision)/(Recall + Precision). Support indicates the number of occurence of the given class in the dataset. The 'macro' is for each label in the dataset, and finds their unweighted mean. The 'weighted' is to calculate metrics for each label, and find their average weighted by support (the number of true instances for each label).

For avoiding the overfit problem, it is necessary to use additional techniques (e.g. cross-validation, regularization). For regularization, c=100 was used in the logistic regression model. This will enable to find a bias-variance tradeoff. For logistic regression, L2 penalty is the default. Lbfgs was used as an optimizer that enables minimal errors and higher accuracy. 'lbfgs' solvers support only l2 penalties.

3.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a widely used machine learning algorithm that can answer both regression and classification problems. The goal of SVM is to define the hyperplane in the space that distinguishes data points into different groups and classifies it. When finding the plane, it has the maximum margin so the distance between data points in different groups is the maximum distance.

confusion matrix:

35	0
0	22

SVM Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	35
1	1.00	1.00	1.00	22
accuracy			1.00	57
macro avg	1.00	1.00	1.00	57
weighted avg	1.00	1.00	1.00	57

Figure 12. Confusion matrix (left) and classification report (right) from Support Vector Machine (SVM) model.

In figure 12, we see that all metrics in the classification report show all 1.00 scores. In the confusion matrix, we don't have any false negatives or false positives.

3.4. Decision tree

Any Algorithm which is not distance based is not affected by feature scaling. Decision tree is a non-parametric and distribution-free algorithm and it's basically rule-based by splitting data and forming the rules. The data input was not scaled for the decision tree. Max_depth was set to 2. Figure 13 represents a decision tree with depth 2.

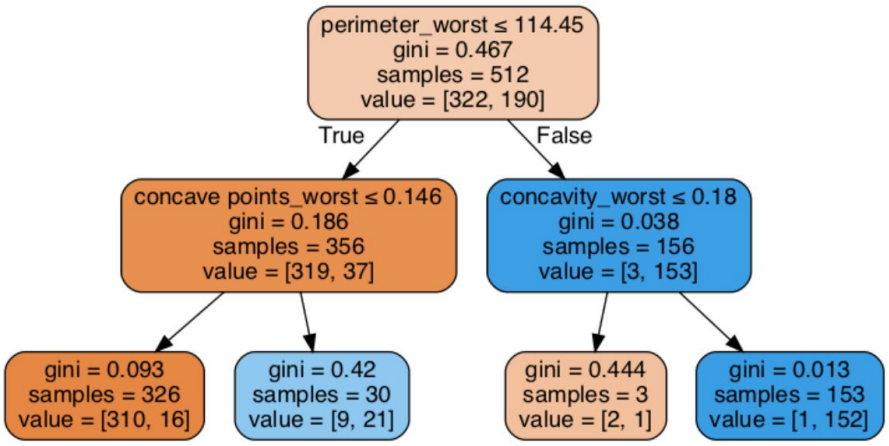

Figure 13. Decision tree modeling of dataset with max_depth =2.

As you can see in figure 13, we can see internal decision-making logic. This tree structure represents features (attribute), branches(a decision rule) and each leaf node (outcome). The best feature (perimeter_worst) was selected using Attribute Selection Measures(ASM) to split the records and make further decision nodes and break into smaller nodes. In the decision tree classifier, max_depth 7 gave the best accuracy (Figure 15). The model performance was tested using a confusion matrix and classification report (Figure 14).

```
confusion matrix:
 [[33  2]
 [ 1 21]]
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.94      0.96        35
           1       0.91      0.95      0.93        22

    accuracy                           0.95        57
   macro avg       0.94      0.95      0.94        57
weighted avg       0.95      0.95      0.95        57
```

Figure 14. Confusion matrix (left) and classification report (right) from Decision tree modeling.

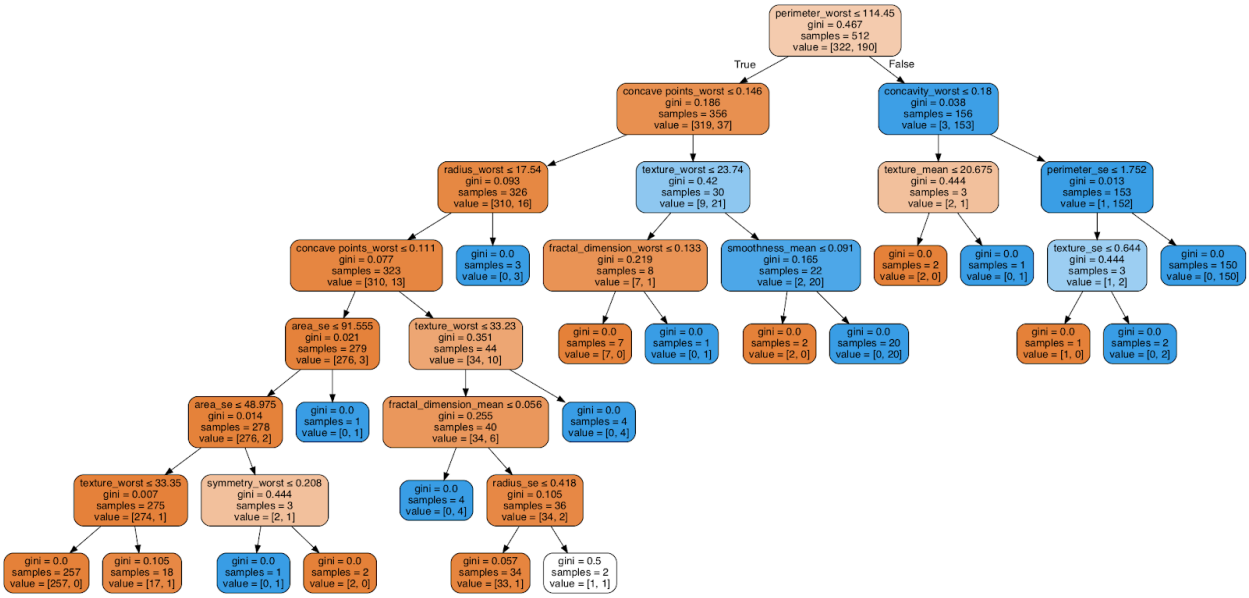

Figure 15. Decision tree modeling of dataset with max_depth =7.

3.5. Random Forest

Random forest is an ensemble method based on bagging (bootstrap aggregation) and consists of many decision trees. This modeling enables each individual tree to randomly sample from the dataset, resulting in different trees. In a random forest classifier, the number of trees in the forest was set at n_estimators=500. Here are the results of model performance.

```
confusion matrix:
 [[33  2]
 [ 1 21]]
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.94      0.96        35
           1       0.91      0.95      0.93        22

    accuracy                           0.95        57
   macro avg       0.94      0.95      0.94        57
weighted avg       0.95      0.95      0.95        57
```

Figure 16. Confusion matrix (left) and classification report (right) from Random Forest.

To find the optimal estimator, the average of Root Mean Square Error (RMSE) was plotted in figure 17.

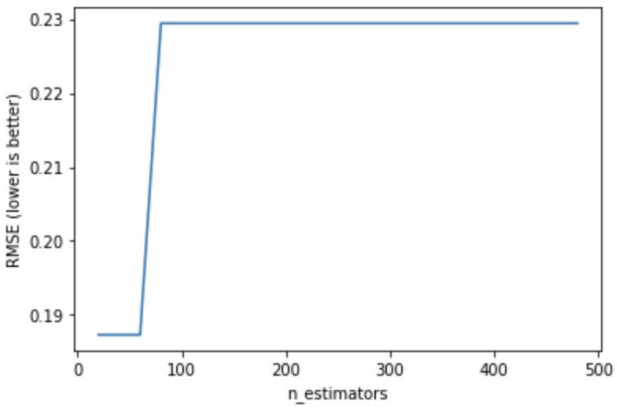

Figure 17. Root Mean Square Error (RMSE) in Y axis and N_estimators in X axis are plotted.

Based on the RMSE, n_estimator = 50 was selected and the model performance is shown in Figure 18.

```
confusion matrix:
 [[32  3]
 [ 0 22]]
```

Classification Report:	precision	recall	f1-score	support
0	1.00	0.91	0.96	35
1	0.88	1.00	0.94	22
accuracy			0.95	57
macro avg	0.94	0.96	0.95	57
weighted avg	0.95	0.95	0.95	57

Figure 18. Confusion matrix (left) and classification report (right) from Random Forest.

Compared to Figure 16 result, the n_estimator =50 model result (Figure 18) shows reduced the number of false negatives in the confusion matrix, and the precision and recall is increased but there is no significant change in the performance results.

3.6. Gradient boosting

Gradient Boosting is a boosting algorithm which produces a prediction model in the form of an ensemble of weak prediction models. By using gradient descent and updating predictions, we can find the values where loss function (MSE) is minimum. The model performance is shown in Figure 19.

```
confusion matrix:
 [[32  3]
 [ 1 21]]
```

Classification Report:	precision	recall	f1-score	support
0	0.97	0.91	0.94	35
1	0.88	0.95	0.91	22
accuracy			0.93	57
macro avg	0.92	0.93	0.93	57
weighted avg	0.93	0.93	0.93	57

Figure 19. Confusion matrix (left) and classification report (right) from Gradient Boosting.

3.7. Deep learning

Deep learning uses “convolutional” neural networks. Neural networks are a set of algorithms that can find patterns modeled like a human brain. It consists of multiple layers and nodes of each layer are clustered, feeding data to multiple nodes. Deep learning is usually used to handle a large dataset. Here, a deep learning model was used to show the model performance and compare it to other machine learning models. In this study, Keras is used to run a deep learning model. Keras is one of deep learning open source Python library. It functions efficiently with numerical computation like Theano and TensorFlow that train neural networks in just a couple of lines of code.

First, Input variables are scaled using StandardScaler. The Keras model consists of sequential models. Connected layers are the dense class. The activation function is used in the activation argument. The rectified linear unit activation function (ReLU) is used on the first two layers and the sigmoid function in the out layer (Figure 20). After we define the Keras model, we have to compile the model (Figure 21). The efficient stochastic gradient descent algorithm “adam” was selected as the optimizer. Since it’s binary classification problems, the loss is defined as “binary_crossentropy”. The metrics argument is defined as the classification accuracy.

```
# Initialising the ANN
classifier = Sequential()

classifier.add(Dense(activation="relu", input_dim=30, units=16, kernel_initializer="unifor
# Adding dropout to prevent overfitting
classifier.add(Dropout(rate=0.1))

# Adding the second hidden layer
classifier.add(Dense(activation="relu", units=16, kernel_initializer="uniform"))
# Adding dropout to prevent overfitting
classifier.add(Dropout(rate=0.1))

# Adding the output layer
classifier.add(Dense(activation="sigmoid", units=1, kernel_initializer="uniform"))
```

Figure 20. Keras sequential model.

```
# Compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Fitting the ANN to the Training set
classifier.fit(X_train, y_train, batch_size=100, epochs=150)
```

Figure 21. Keras compile model.

The keras model was fit with a train dataset(Figure 21). Epoch indicates one pass through all rows in the training dataset. Batch indicates one or more samples within an epoch before weights are updated.

```
confusion matrix:
 [[35  0]
 [ 0 22]]
```

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        35
           1       1.00      1.00      1.00        22

    accuracy                           1.00        57
   macro avg       1.00      1.00      1.00        57
weighted avg       1.00      1.00      1.00        57
```

Figure 22. Keras model performance.

In figure 22, model performance results indicate that all metrics in the classification report show 1.00 scores. In the confusion matrix, we don't have any false negatives or false positives.

4. Model summary and recommendations

In this study, 6 different machine learning algorithms are compared for the performance. In figure 23, both support vector machines(SVM) and deep learning show all 1.00 scores.

	accuracy	precision	recall	f1-score
1. Logistic regression	0.98	1.00	0.95	0.98
2. Support vector machine	1.00	1.00	1.00	1.00
3. Decision Tree	0.95	0.91	0.95	0.93
4. Random forest	0.95	0.88	1.00	0.94
5. Gradient Boosting	0.93	0.88	0.95	0.91
6. Deep learning	1.00	1.00	1.00	1.00

Figure 23. Prediction of malignancy: comparison of models.

5. Dimension reduction

Dimension reduction is the process of reducing several features in the dataset into new independent variables. In this study, PCA (Principal Component Analysis) is used for dimension reduction. PCA is the most common method of linear transformation. Prior to PCA analysis,

input data was normalized using StandardScaler. To find features that show the maximum variance, two features will be the principal components and plotted in the graph (see Figure 24).

Figure 24. The scatter plot of breast cancer prediction classes based on 2 principal components.

In scatter plot, we see that malignant class is more spread out compared to benign class, meaning that there are more variations in features in malignant class. It is hard to get the exact features used for the first and second principal component since these components were calculated from some mixture of the original features and it is an unsupervised method. PCA analysis is efficient when we handle a large data-set with many features and reduce the dimension of features.

6. Conclusions

Here we explored the breast cancer dataset with 31 features obtained from patients' biopsy samples and classify them into benign and malignant results. Using performance evaluation metrics, we found that both support vector machines and deep learning algorithms predicted these classes with 100% accuracy. The limitation is that we assume that each feature is independent and this makes it easier to apply different models. Further analysis with different data sources will be helpful for evaluating these models in depth.