Predicting Breast Cancer Using Machine Learning: An Analysis of Logistic Regression, Random Forest and K-Nearest Neighbor Algorithms

Norhaan Saadawi
Faculty of Computing, Mathematics, and Data Science
Coventry University
Coventry, England
saadawin@coventry.ac.uk

***Abstract*---In this paper, prediction of breast cancer using machine learning techniques such as logistic regression, random forest, and k-nearest neighbors were analysed. These algorithms were tested using the Wisconsin breast cancer (diagnostic) data set. The algorithms were tested for accuracy and efficiency based on precision, sensitivity, specificity, F1 scoring, receiver operating characteristic (ROC) curves, confusion matrices, time it took to fit the model and accuracy. The results showed that logistic regression performed the best followed by k-nearest neighbors and lastly random forest. All algorithms showed to be very effective and accurate in correctly predicting breast cancer and provide great insights for the healthcare and artificial intelligence field to be further researched.**

I. Introduction

Breast cancer is a type of cancer which involves abnormal breast cells growing uncontrollably forming malignant tumours [1]. These tumours, if not properly treated, can spread to other parts in the body and become deadly [1].

In 2022, breast cancer was the cause of 670,000 deaths worldwide [1]. In 2022, it was also the most common cancer in women in 157 countries out of 185 [1]. Today breast cancer remains to be the most common cancer in women in the United Kingdom [2]. This information is very alarming and additional measures need to be taken to reduce deaths caused by breast cancer.

Tumours in the body are either classified as benign or malignant. A benign tumour is not cancerous while a malignant tumour is cancerous [3]. Due to this, the diagnosis process for breast cancer can involve many tests which can include imaging tests, laboratory tests, or biopsy [3]. Laboratory tests show practitioners the amounts of certain substances in the body which can be a sign of cancer [4]. These tests are used alongside other tests to confirm if a person has cancer or not as they cannot confirm if a person has cancer only based on substance levels in the body [4]. Imaging tests show whether a tumour is present or not in the body but do not show if the tumour is cancerous or not [4]. Lastly, a biopsy can be done to accurately confirm if a tumour is cancerous or not [4].

II. Literature Review

The early detection of breast cancer is very important in reducing deaths and impairments from the disease. Machine learning can assist along with the other screening tests that are available to determine whether an individual has breast cancer or not. Over the years, using machine learning to predict health outcomes has increased in popularity and many studies have been done to showcase the relationship between machine learning and the prediction of breast cancer. Different varieties of data can be used to predict breast cancer using machine learning. This includes features such as mammography images, phenotypes, genomes, or cell nucleus attributes. However, the focus of this study is based on detection using cell nucleus attributes so this will be the scope of the literature to be reviewed.

In the literature it has been found that Amrane and colleagues compared K-nearest neighbors (KNN) and naive bayes (NB) classifiers in predicting whether a breast cancer tumour is benign or malignant [5]. The study was done on 683 instances each containing 9 features [5]. The features included clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitosis [5]. The results found that KNN had 97.51% accuracy compared to NB which has 96.19% accuracy [5]. The only limitation that was found in this research was that KNN is likely to have large running time if the dataset were larger but for this sample it was more efficient that NB [5]. This study provides a good example of how the KNN algorithm can provide high accuracy for detecting breast cancer when using cell attributes as features [5].

Umer and colleagues used an ensemble learning-based voting classifier and compared it to other machine learning algorithms on the Wisconsin Breast Cancer dataset [6]. The ensemble learning-based voting classifier combined stochastic gradient descent and the logistic regression classifier to aid in the identification of cancerous tumours [6]. The results of this study showed that when the voting ensemble model containing stochastic gradient descent and logistic regression was compared to random forest, KNN,

support vector machines, and decision trees, it performed the best. The voting ensemble model achieved an accuracy score of 77.2% [6]. An advantage of this study is that they tested the features from the original dataset without feature engineering then did feature engineering using the convoluted neural network method and compared the results to each other [6]. The accuracy after carrying out the feature engineering on the features rose to 100% for the ensemble learning-based voting classifier. Thus, providing a stepping stone for future research to incorporate this method when doing predictions with machine learning on healthcare data [6].

Obaid and colleagues compared support vector machines, decision trees, and KNN in classifying tumours for the Wisconsin breast cancer dataset [7]. It was found that support vector machines performed the best compared to the other models giving an accuracy score of 98.1% for the quadratic kernel, 97.9% for the linear kernel, and 97% for the cubic kernel [7]. This study provides a good example of how the support vector machines can provide high accuracy for detecting breast cancer when using cell attributes as features [7]. The advantages of this study are that they investigated each of the three machine learning models in great depth [7]. They investigated the KNN classifier for fine, medium and coarse KNN types and decision trees using complex, medium, and simple trees, and support vector machines for quadratic, linear, and cubic kernels [7].

In another study, Asri and colleagues compared support vector machines, naive bayes, KNN, and decision trees on the Wisconsin breast cancer dataset [8]. The results of their research found that support vector machines performed the best with an accuracy of 97.13% which overlaps with the results found by Obaid and colleagues [8]. This study further supports the use of support vector machines algorithm for the prediction of breast cancer when using cell attributes as features.

Islam and colleagues investigated decision trees, naive bayes, logistic regression, random forest, and extreme gradient boosting on data from hospitals in Bangladesh [9]. The features used in the study were cell attributes such as radius, perimeter, texture, smoothness, and area and they were used to predict the diagnosis being either benign or malignant for breast cancer [9]. The results found that random forest and extreme gradient boost had the highest accuracy of 94% [9]. This study, although working with a similar dataset to the Wisconsin breast cancer dataset, got different results for the evaluation of the different machine learning models when compared to the other literature [9]. A limitation of this study is that the data was collected during the coronavirus pandemic so it was difficult to get a large sample [9]. The researchers suggest that if the dataset used were larger, the results would be more effective and accurate for adding to the field of cancer detection [9].

III. Problem and Data Set

Machine learning comes into this issue by helping pathologists in reviewing biopsy results and classifying cells are benign or malignant in tumours. This technique is used in healthcare and this study will aim to review three different classification techniques that can be used for this issue and analyse their performance by comparing accuracy results.

The dataset used is the breast cancer Wisconsin (diagnostic) dataset from UCI machine learning repository. The dataset contains 569 instances, 30 numeric features and no missing values. The target variable is diagnosis and it is categorical with two outcomes to classify a tumour, either benign or malignant. The dataset contains 212 benign samples and 357 malignant samples. Each instance is a cell nucleus and each feature are the attributes of the cell nucleus that help pathologists determine whether the cell is cancerous or not. There are 10 attributes that are being looked at in this data set with each containing the mean, standard error, and worst measures to produce a total of 30 features. A summary table of the dataset feature names and descriptions is presented in table 1 in Appendix B.

IV. Methods

A. K-Nearest Neighbors

KNN is a supervised learning method that can work on both classification and regression problems [10]. In this method, a number of nearest neighbours and distance is chosen before starting [10]. The number of nearest neighbours refers to how many data points will be looked at when predicting a class [10]. The distance determines the similarity between the test and training data points [10]. This method then works by grouping the data points that are closest to each other and making them ‘neighbours [10]’. Majority voting is then done for classification problems among the neighbours to determine the predicted class [10]. In regression problems, the average is taken among the neighbours to determine the predicted class [10]. Advantages of KNN are that it is easily adaptable, uses few hyperparameters, and it is easy to implement [10].

B. Random Forest

Random forest is a supervised learning method that can work on both regression and classification problems [11]. This method works by creating many decision trees and combining them together [11],[12]. Each decision tree that is created is made using a random subset from the data and measures a random subset of features to maximise class separation when working on classification problems, and minimise variance when working on regression problems [11],[12]. The method then does majority voting by summing all the results from all the decision trees for classification problems to determine the predicted class [11],[12]. In order to determine the predicted class for regression problems, an average is taken from all the results from the decision trees [11],[12]. Advantages of this method are that it reduces the risk of overfitting and easily measures feature importance

[11]. A detailed image of how the random forest process works for classification problems is presented in figure 1.

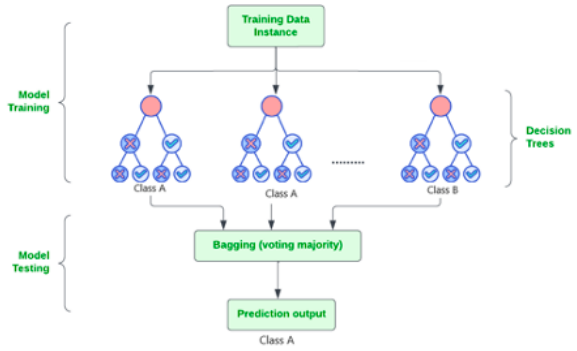

Figure 1. Random forest for classification problems [13]

C. *Randomized searchcv*

This is a hyperparameter tuning technique that optimises the parameters to be used in a chosen classifier [14]. This method chooses at random a fixed amount of parameters to test from the specified parameters and gives a score to each [14]. The parameters with the highest score are returned to the user [14].

D. *GridSearch*

This hyperparameter tuning technique tests out every combination of parameters given in the list of parameters to try and returns the parameters with the best score to the user [15].

E. *Logistic Regression*

Logistic regression is a supervised learning method that works to determine classifications between different categories [16]. The three types of logistic regression are binary, multinomial and ordinal [17]. Binary was used in this study as it is used when the predicted outcome is binary. Multinomial is used when the predicted outcome can be chosen from three or more classes [17]. Lastly, ordinal regression is used when the predicted outcome has a predetermined order [17]. A limitation of this method is that it can be prone to overfitting when there are many features used in the model [17]. An advantage of this model is that it is widely used in the prediction of diseases and has been shown to be successful in much of the literature [17].

V. Experimental Setup

Preprocessing was done using Spyder (Python 3.11) and Microsoft Excel version 16.86 on a 2017 13 inch Macbook Air. The dataset was downloaded from the UCI website and converted to an excel file in order to be opened, have the columns separated and the overall file converted to a csv file to be used in python. Once in Python, the 'ID number' column was removed as it would not be used as a feature in the machine learning prediction algorithms. The columns were named in Python and the categories of the predicted outcome were changed to binary values 0 for malignant and 1 for benign. Lastly, the data was split into two groups to separate the features from the output variable and normalisation was done to the features. The code inputs and outputs used in this paper can be found in Appendix A.

VI. Results

All algorithms were carried out using Spyder (Python 3.11) on a 2017 13 inch Macbook Air. The random forest classifier had hyperparameters optimised using randomized search and the KNN classifier had hyperparameters optimised using grid search both from the scikit-learn library in Python. The training set consisted of 80% of the data and the test set consisted of 20% of the data. Each model was trained using all 30 features and further information regarding the trial of feature selection and extraction methods can be found in the discussion section. Each model's efficiency was evaluated based on confusion matrices, accuracy scores, precision scores, F1 scores, sensitivity and specificity scores, time it takes to train the model, and ROC curves. All were calculated using Python scikit-learn library except for sensitivity and specificity where formulas are found in figure 2 along with sample calculations for the logistic regression algorithm in figure 3. The summary of all the evaluation metrics for each of the models are presented in table 2.

Algorithm	Accuracy	Sensitivity	Specificity	Precision	F1 score	Time to fit model
Logistic Regression	0.982	1	0.947	0.974	0.987	0.01
KNN	0.973	0.986	0.947	0.974	0.980	0.0009
Random Forest	0.956	0.960	0.947	0.973	0.966	0.322

Table 2. Summary of evaluation metrics

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

Figure 2. Equations used to calculate sensitivity and specificity for each algorithm [18]

Sample Calculations for Logistic Regression:

$$\text{Sensitivity} = \frac{76}{76+0}$$
$$\text{Sensitivity} = 1$$

$$\text{Specificity} = \frac{36}{36+2}$$
$$\text{Specificity} = 0.947$$

Figure 3. Sample equations for sensitivity and specificity for logistic regression

A. *Logistic Regression*

Logistic Regression was done in Python using the scikit-learn library. The results for the confusion matrix are presented in figure 4 and the results for the ROC curve are presented in figure 5 . The accuracy scores, precision scores, F1 scores, sensitivity and specificity scores, and time it takes to train the model are presented in table 2.

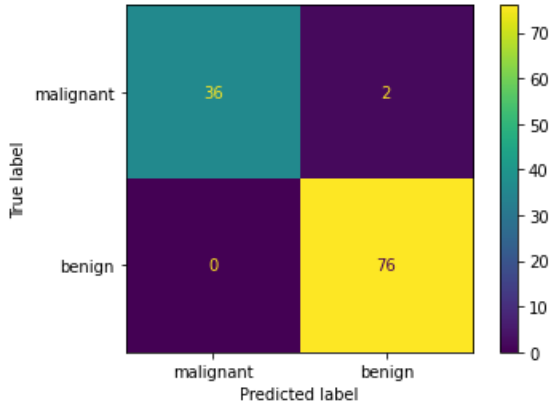

Figure 4. Confusion matrix for logistic regression

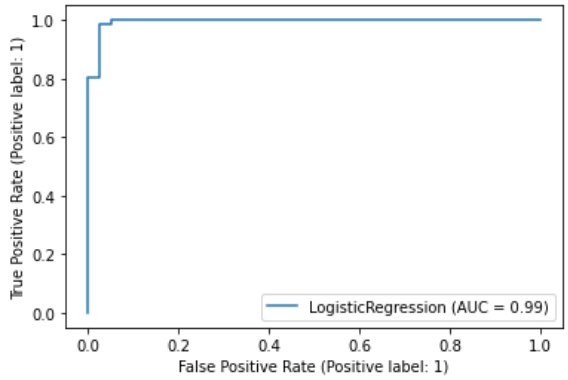

Figure 5. ROC curve for logistic regression

B. *KNN*

KNN was done in Python using the scikit-learn library. The optimal hyperparameters that were used for the chosen dataset were 3 neighbours and the uniform weight both provided by randomized search. The results for the confusion matrix are presented in figure 6 and the results for the ROC curve are presented in figure 7. The accuracy scores, precision scores, F1 scores, sensitivity and specificity scores, and time it takes to train the model are presented in table 2.

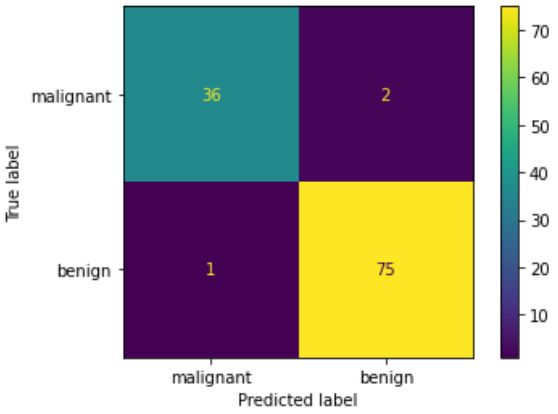

Figure 6. Confusion matrix for KNN

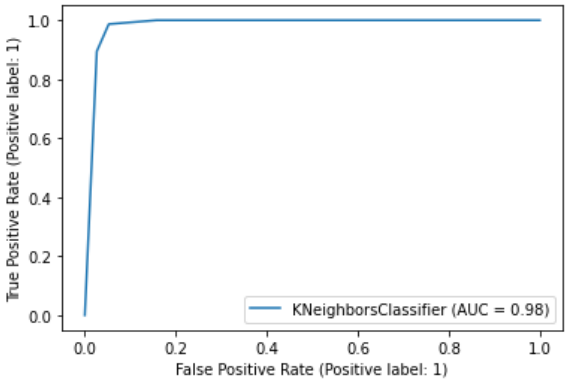

Figure 7. ROC curve for KNN

C. *Random Forest*

Random Forest was done in Python using the scikit-learn library. The optimal hyperparameters that were used for the chosen dataset were 322 estimators and the entropy criterion both provided by randomized search. The results for the confusion matrix are presented in figure 8 and the results for the ROC curve are presented in figure 9. The accuracy scores, precision scores, F1 scores, sensitivity and specificity scores, and time it takes to train the model are presented in table 2.

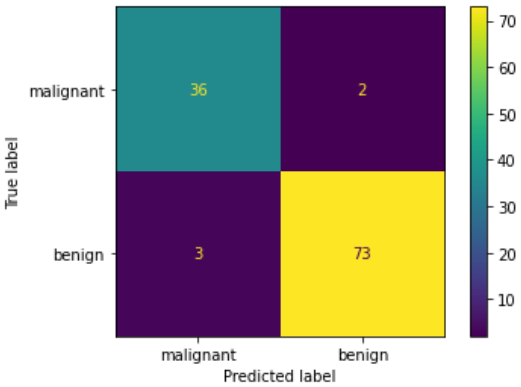

Figure 8. Confusion matrix for random forest

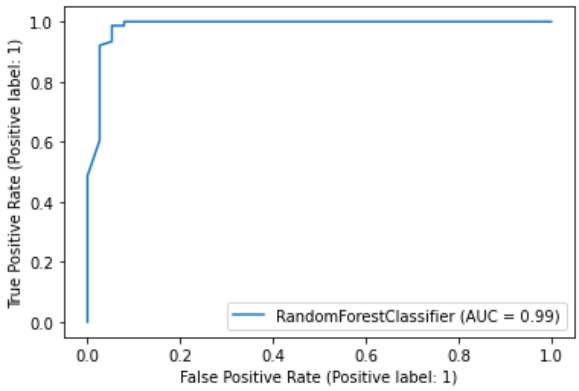

Figure 9. Roc curve for random forest

VII. Discussion

The results in table 2 show that all models performed the same in specificity and both logistic regression and KNN performed the same in precision. Logistic regression performed the best in accuracy, sensitivity, and F1 scoring. Random forest performed the worst in accuracy, sensitivity, precision, F1 scoring, and time to fit the model. KNN took the fastest time to fit the model, while random forest took the longest time. Seen in the ROC curves in figures 5,7 and 9, random forest and logistic regression have the same area under the curve and KNN was below by only 0.01. All the models performed excellently with area under the curve all over 0.9 as well as all evaluation metrics over 0.9. The confusion matrices in figures 4,6, and 8 all show very high values for the true positives and true negatives and very low values for the false positives and false negatives, further showing that the models performed very well. Overall, logistic regression performed the best, followed by KNN, and last was random forest.

Although the models all performed extremely well using all 30 features, feature selection was tried using sequential forward selection from the mlxtend library for backwards and forwards selection along with select k best from the scikit-learn library. The performance of the models however became worse and decreased when these techniques were tried. Similarly, principal component analysis (PCA) was tried for dimension reduction keeping 95% of the variance in 10 components using PCA from the scikit-learn library and the performance of the models remained the same. There is no change between using all 30 features in the models or doing dimension reduction with PCA.

VIII. Social, Legal, and Ethical Considerations

Ethical issues to consider regarding using machine learning to predict breast cancer is bias, and patient consent. Bias can occur during the data collection stage if data is not collected from different patient populations, if lots of data is removed during the data cleaning and preparation stage, or due to the type of model used and the evaluations used on the model. Patient consent is also very important to receive before using the patient's information to input into the machine learning model. The laws regarding patient consent are different worldwide so researchers and healthcare practitioners wanting to implement this model in healthcare would need to do additional research specific to their location to determine how to receive patient consent for this procedure. Legal issues to consider can be liability and privacy. Liability needs to be considered as the machine learning model could provide incorrect predictions and if not double checked by a healthcare professional this can harm patients. Patients can be harmed if they do not receive appropriate treatment in time or if they receive incorrect treatment or treatment that is not needed. Thus, it is important to know who is legally responsible if things go wrong with the disease prediction and what measures are in place to protect the healthcare professionals from being sued. Privacy of patient data, again is different worldwide but in many areas the main goal is keeping patient information private so ensuring that the machine learning model can meet this is essential to its application in healthcare. Social issues to consider are access to the model and trust. If the model is implemented in healthcare it should be available in all populations not only specific areas. Additionally, healthcare professionals and the public should be informed about the machine learning process to ensure they trust and accept the model to perform important predictions.

IX. Conclusions and Future Research

In conclusion, logistic regression performed the best followed by KNN and random forest. Overall, all the algorithms performed very well and provide promising results for future research to be done to implement machine learning models to help in disease prediction. Future research could also aim to use larger datasets and different datasets that include data from different populations. The data in the Wisconsin breast cancer (diagnostic) dataset does not mention which populations the data is taken from but if there is additional information regarding the population it may help to learn more about if cell attributes differ between different populations. Additionally, since the dataset used is very old, newer research could use data from recent hospital samples for more accurate data to see if cell attributes have changed over time due to environmental or diet changes over the years. This can improve machine learning accuracy and provide new insights in the healthcare and artificial intelligence field that have not been researched fully.

X. References

1. World Health Organization, "Breast cancer," World Health Organization, Mar. 13, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/breast-cancer

2. NHS, "What is breast cancer in women?," NHS choices, Mar. 4, 2024. [Online]. Available: https://www.nhs.uk/conditions/breast-cancer-in-women/what-is-breast-cancer-in-women/.

3. Cancer Research UK, "How cancers grow," Cancer Research UK, Jun. 11, 2024. [Online]. Available:

https://www.cancerresearchuk.org/about-cancer/what-is-cancer/how-cancers-grow.

4. National Cancer Institute, "Tests and procedures used to diagnose cancer," Tests and Procedures Used to Diagnose Cancer - NCI, Jan. 17, 2023. [Online]. Available: https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis#lab-tests-used-to-diagnose-cancer.

5. M. Amrane, S. Oukid, I. Gagaoua, and T. Ensari, "Breast cancer classification using machine learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, Apr. 2018, pp. 1-4.

6. M. Umer et al., "Breast cancer detection using convoluted features and ensemble machine learning algorithm," *Cancers*, vol. 14, no. 23, p. 6015, 2022.

7. O. I. Obaid et al., "Evaluating the performance of machine learning techniques in the classification of Wisconsin Breast Cancer."

8. H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, "Using machine learning algorithms for breast cancer risk prediction and diagnosis," *Procedia Computer Science*, vol. 83, pp. 1064-1069, 2016.

9. T. Islam et al., "Machine learning approaches to predict breast cancer: Bangladesh perspective," in *International Conference on Ubiquitous Computing and Intelligent Information Systems*, Singapore: Springer Nature Singapore, Apr. 2021, pp. 291-305.

10. T. Srivastava, "Guide to K-nearest neighbors algorithm in machine learning," Analytics Vidhya, May 21, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/#:~:text=It%20works%20by%20finding%20the,the%20field%20of%20artificial%20intelligence

11. IBM, "What is Random Forest?," IBM, Oct. 20, 2021. [Online]. Available: https://www.ibm.com/topics/random-forest.

12. N. Donges, "Random Forest: A complete guide for machine learning," Built In, Mar. 8, 2024. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm

13. GeeksforGeeks, "Random Forest algorithm in machine learning," GeeksforGeeks, Jul. 12, 2024. [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/.

14. Scikit learn, "RANDOMIZEDSEARCHCV," Scikit learn, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html.

15. Scikit learn, "GRIDSEARCHCV," Scikit learn, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

16. IBM, "What is logistic regression?," IBM, Aug. 16, 2021. [Online]. Available: https://www.ibm.com/topics/logistic-regression.

17. A. Saini, "What is logistic regression?," Analytics Vidhya, Jun. 19, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/.

18. A. Mitrani, "Evaluating categorical models II: Sensitivity and specificity," Medium, Dec. 6, 2019. [Online]. Available: https://towardsdatascience.com/evaluating-categorical-models-ii-sensitivity-and-specificity-e181e573cff8.

19. W. Wolberg, O. Mangasarian, N. Street, and W. Street, "Breast cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, Oct. 31, 1995. [Online]. Available: https://archive.ics

XI. Appendix A

Code Input:

```
from sklearn.metrics import RocCurveDisplay

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay

from sklearn.metrics import confusion_matrix

from sklearn.metrics import precision_score, f1_score

from sklearn.model_selection import GridSearchCV

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import RandomizedSearchCV

import time


#Opening the file and naming the columns

data = pd.read_csv('cancerdata.csv', names = ['ID number', 'Diagnosis', 'Radius Mean','Texture Mean','Perimeter Mean','Area Mean','Smoothness Mean','Compactness Mean','Concavity Mean','Concave Points Mean','Symmetry Mean','Fractal Dimension mean' ,'Radius SE','Texture SE','Perimeter SE','Area SE','Smoothness SE','Compactness SE','Concavity SE','Concave Points SE','Symmetry SE','Fractal Dimension SE' ,'Radius Worst','Texture Worst','Perimeter Worst','Area Worst','Smoothness Worst','Compactness Worst','Concavity Worst','Concave Points Worst','Symmetry Worst','Fractal Dimension Worst' ])


#Preprocessing and data cleaning

data = data.drop('ID number', axis = 1)

data['Diagnosis'].replace(['M','B'],[0,1], inplace = True)

X = data.drop('Diagnosis', axis = 1)

Y = data['Diagnosis']


#Normalizing the data
```

```
X = MinMaxScaler().fit_transform(X)

#Splitting the data

x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size = 0.2)


#Applying logistic regression and calculating evaluation metrics, curve, matrix, and calculating fit time

lr_clf = LogisticRegression()

start = time.time()

lr_clf.fit(x_train,y_train)

end = time.time()

print('time to fit the model:',end-start)

y_pred_lr = lr_clf.predict(x_test)

print('accuracy score is:',accuracy_score(y_test,y_pred_lr))

lr_cm = confusion_matrix(y_test,y_pred_lr)

print(lr_cm)

lbls = ['malignant','benign']

ConfusionMatrixDisplay(lr_cm,display_labels= lbls).plot()

RocCurveDisplay.from_estimator(lr_clf,x_test,y_test)

plt.show()

precision = precision_score(y_test,y_pred_lr)

print('precision is',precision)

f1 = f1_score(y_test, y_pred_lr)

print('f1 score is',f1)


#Applying KNN,grid search and calculating evaluation metrics, curve, matrix and calculating fit time

parameters = {'n_neighbors' : range(1,30), 'weights': ['uniform', 'distance']}

Grid_search = GridSearchCV(estimator = KNeighborsClassifier(), param_grid = parameters,scoring = 'accuracy',
return_train_score= True)

Grid_search.fit(x_train,y_train)

print(Grid_search.best_params_)

knn_clf = KNeighborsClassifier(n_neighbors=3)

start = time.time()
```

```
knn_clf.fit(x_train,y_train)

end = time.time()

print('time to fit the model:',end-start)

y_pred_knn = knn_clf.predict(x_test)

print('accuracy score is:',accuracy_score(y_test,y_pred_knn))


knn_cm = confusion_matrix(y_test,y_pred_knn)

print(knn_cm)

ConfusionMatrixDisplay(knn_cm,display_labels=lbls).plot()

RocCurveDisplay.from_estimator(knn_clf,x_test,y_test)

plt.show()

precision = precision_score(y_test,y_pred_knn)

print('precision is',precision)

f1 = f1_score(y_test, y_pred_knn)

print('f1 score is',f1)


#Applying random forest, randomized search and calculating evaluation metrics, curve, matrix and calculating fit time

parameters2 = {'n_estimators' : range(1,1000), 'criterion': ['gini','entropy']}

rand_search = RandomizedSearchCV(estimator = RandomForestClassifier(), param_distributions=parameters2, scoring =
'accuracy', return_train_score=True)

rand_search.fit(x_train, y_train)

print(rand_search.best_params_)

rf_clf = RandomForestClassifier(n_estimators=332, criterion='entropy')

start = time.time()

rf_clf.fit(x_train,y_train)

end = time.time()

print('time to fit the model:',end-start)

y_pred_rf = rf_clf.predict(x_test)

print('accuracy score is:',accuracy_score(y_test,y_pred_rf))

rf_cm = confusion_matrix(y_test, y_pred_rf)

print(rf_cm)
```

```
ConfusionMatrixDisplay(rf_cm,display_labels= lbls).plot()

RocCurveDisplay.from_estimator(rf_clf,x_test,y_test)

plt.show()

precision = precision_score(y_test,y_pred_rf)

print('precision is',precision)

f1 = f1_score(y_test, y_pred_rf)

print('f1 score is',f1)
```

Code Output:

Logistic regression:

```
time to fit the model: 0.010488033294677734
accuracy score is: 0.9824561403508771
[[36  2]
 [ 0 76]]
precision is 0.9743589743589743
f1 score is 0.9870129870129869
```

KNN:

```
{'n_neighbors': 3, 'weights': 'uniform'}
time to fit the model: 0.0009720325469970703
accuracy score is: 0.9736842105263158
[[36  2]
 [ 1 75]]
precision is 0.974025974025974
f1 score is 0.9803921568627451
```

Random forest:

```
{'n_estimators': 332, 'criterion': 'entropy'}
```

```
time to fit the model: 0.32272791862487793
accuracy score is: 0.956140350877193
[[36  2]
 [ 3 73]]
precision is 0.9733333333333334
f1 score is 0.9668874172185431
```

XII. Appendix B

Table 1. Dataset features and descriptions [6],[19]

Feature	Description
Mean radius	Average distance from the centre to perimeter points
Mean texture	Average grey-scale value standard deviation
Mean perimeter	Average core tumour size
Mean area	Average area of the cell nucleus
Mean smoothness	Average variation in radius lengths
Mean compactness	Average perimeter ^2/area-1
Mean concavity	Average severity of concave areas
Mean concave points	Average number of concave points
Mean symmetry	Average symmetry of the cell nucleus
Mean fractal dimension	Average ‘coastline approximation’ - 1
Standard error radius	Standard error for the distance from the centre to perimeter points
Standard error texture	Standard error for grey-scale value standard deviation
Standard error perimeter	Standard error for tumour size
Standard error area	Standard error for the area of the cell nucleus
Standard error smoothness	Standard error for the variation in radius lengths
Standard error compactness	Standard error of perimeter ^2/area-1
Standard error concavity	Standard error for the severity of concave areas
Standard error concave points	Standard error for the number of concave points
Standard error symmetry	Standard error for symmetry of the cell nucleus
Standard error fractal dimension	Standard error for ‘coastline approximation’ - 1
Worst radius	Largest average distance from the centre to perimeter points
Worst texture	Largest average grey-scale value standard deviation
Worst perimeter	Largest average core tumour size
Worst area	Largest average area of the cell nucleus
Worst smoothness	Largest average variation in radius lengths
Worst compactness	Largest average perimeter ^2/area-1
Worst concavity	Largest average severity of concave areas
Worst concave points	Largest average number of concave points
Worst symmetry	Largest average symmetry of the cell nucleus
Worst fractal dimension	Largest average ‘coastline approximation’ - 1