Analysis and Prediction of PIMA Indian Diabetes Dataset Using Eight Classifier Techniques

Rui Wang
School of Computer Science
University of Nottingham
Nottingham, UK
psxrw10@nottingham.ac.uk

Yue Xue
School of Computer Science
University of Nottingham
Nottingham, UK
psxyx15@nottingham.ac.uk

***Abstract*—Diabetes is a serious chronic disease that kills over a million people every year. In addition, diabetes of all types can lead to complications in many parts of the body and can increase the overall risk of dying prematurely. Therefore, diagnosing diabetes as early as possible has great medical significance. Compared to traditional methods of diabetes diagnosis, using algorithms or models, such as ANN, SVM, decision trees, etc., the efficiency and accuracy of diagnosis can be significantly improved. Therefore, this thesis compares several algorithms or models and then implements some of them based on the R language to diagnose whether a patient has diabetes. This research has chosen the Pima Indians Diabetes Database dataset, which contains 9 attributes with 768 samples. Furthermore, accuracy, precision, recall, F1-Measures and ROC, AUC criteria are used to evaluate performance. Eventually, we concluded that the testing accuracy and AUC of the eight methods. The best performer among the eight models was Logistic Regression with accuracy and AUC of 0.792 and 0.855.**

Keywords—Diabetes Mellitus, Data Mining, Classification, Decision Tree, Artificial Neural Networks, Pima Indians Diabetes Database

1. Introduction

1.1 Background

Diabetes Mellitus (DM) is one of the major health issues in the world [1,2]. World Health Organization (2019) estimates that 422 million adults were suffering from diabetes in 2014 in comparison to 108 million 34 years ago. The popularity of diabetes has doubled globally since 1980 and the amount of diabetes patients has risen from 4.7% to 8.5% in the adult population [3]. If it cannot be found early and controlled early, the health consequences of diabetes are very serious. Diabetes can damage the heart, blood vessels, kidneys, eyes and nerves. For example, the amputation rate of the lower limbs of diabetic patients is 10 to 20 times higher than others. For poor people around the world, the cost of controlling diabetes would be catastrophic and drag the family below the poverty line. This costs will also consume the sanitary budget of a country and weaken the national economy. The WHO estimates that the direct medical expenses for DM would be 830 billion US dollars each year [3]. Diabetes would not only hurt the body, but also kill the life. DM is known as a silent killer in the medical area [4]. Around 1.5 million people die from DM every year and high blood glucose would causes another 2.2 million death because it exacerbates the risk of cardiovascular diseases [3]. Many such diseases can be prevented if the DM diseases are predicted and controlled in advance.

1.2 DM Types

There are basically four types of diabetes mellitus: type 1 diabetes, type 2 diabetes, gestational diabetes, and specific types of diabetes. Type 1 diabetes (also known as insulin-dependent diabetes), which accounts for about 5% to 10% of all diabetic patients usually occurs in children and adolescents, but can occur at any age, even in the 80s and 90s. The etiology is that the patients' β cells are damaged by cell-mediated autoimmunity and cannot synthesize and secrete insulin by themselves. Insulin injection is needed for this kind of patients so as to ensure the normal level of their blood glucoses. Type 2 diabetes accounts for about 90% of the total number of diabetic patients, and the age of onset is mostly after the age of 35. About 60% of people with type 2 diabetes are overweight or obese. Type 2 diabetes has obvious familial inheritance. Gestational diabetes mellitus means the diabetes that is usually detected in the middle or later time of pregnancy, while DM in pregnancy means Type 1 or type 2 diabetes first diagnosed during pregnancy. There are also some other specific types of DM [3,5].

1.3 Traditional Diagnostic Methods

Four diagnostic tests for diabetes are currently recommended, including measurement of fasting plasma glucose, 2-hour post-load plasma glucose after a 75 gram oral glucose tolerance test (OGTT), Hemoglobin A1C (HbA1c) test, and a random blood glucose when symptoms of DM is detected. People with fasting plasma glucose values over 7.0 mmol/L (126 mg/dl), or 2-h post-load plasma glucose greater than 11.1 mmol/L (200 mg/dl), or HbA1c over 6.5% (48 mmol/mol), or a random blood glucose over 11.1 mmol/L (200 mg/dl) in the presence of symptoms of DM are considered to have diabetes. Repeated testing is recommended to confirm the diagnosis [3].

1.4 Diagnostics using Machine Learning

Compared to traditional diagnostic methods, machine Learning makes diagnosis progress easier and deterministic. So in need of early detection and predoction of diabetes. Therefore machine learning methods are real exemplary, in this aspect for accurate predictions of diabetes data which may help the patient in more consolidated way. According to Hasan et al. (2020), diabetes could not be cured but could be controlled well if it was predicted in advanced [6].

1.5 Dataset Description

This research utilise the Pima Indian diabetes datasets (PIDD) created by Smith et al. (1988) [7]. Pima Indians from Arizona have a high risk of non-insulin-dependent diabetes mellitus (DM), therefore the National Institute of Diabetes and Digestive and Kidney Disease of the United States has conducted physical examinations for the population every two years since 1965. The PIDD consists of one outcome and eight input variables including pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function and age. These variables are considered as important risk points for DM in the population. As for the outcome, 1 stands for the DM positive while 0 means DM negative. There are totally 768 instances in the PIDD. This dataset is taken from Kaggle (2016), and it is considered as the most famous dataset for training and testing classification techniques [1, 8].

TABLE I. VARIABLES IN PIDD (KAGGLE, 2016)

Number	*Attribute*	*Description*
1	Pregnancies	Independent variable; Times of pregnancy
2	Glucose	Independent variable; Plasma glucose concentration in an oral glucose tolerance test
3	BloodPressure	Independent variable; Diastolic blood pressure (mm Hg)
4	SkinThickness	Independent variable; Triceps skin fold thickness (mm)
5	Insulin	Independent variable; 2-Hour serum insulin (mu U/ml)
6	BMI	Independent variable; Body mass index (kg/m^2)
7	DiabetesPedigreeFunction	Independent variable; Genetic relationship of diabetes risk
8	Age	Independent variable
9	Outcome	Dependent/Target variable; Diabetes positive = 1, diabetes negative = 0

2 LITERATURE REVIEWS

There has been considerable researches regarding the DM and PIDD. This chapter will discuss some researches regarding the prediction models on DM with PIDD.

Smith et al. (1988) create PIDD and investigate the performance of a neural network model, ADAP on DM diagnose [7]. 576 cases are used for training and 192 cases are used for forecasting. This research use sensitivity, specificity, and a receiver operating characteristic (ROC) curve to evaluate the performance of the model.

Sivanesan and Dhivya (2017) use classification technique to compare the diagnoses of DM. Classification could predict the result with given inputs [1]. This research employs J48 decision tree algorithm to classify the PIDD data and predict whether the person is diabetes positive or not. The dataset utilise three resampling methods, which are training set method, 10 fold cross validation method and percentage split model method (65% and 35%).

Soltani and Jafarian (2016) use probabilistic artificial neural networks (PNN) in MATLAB to maximise the training and testing performance of PIDD [2]. 90% of the sample are used for training and 10% of the dataset are used for testing. This research achieves 89.56% and 81.49% for the training accuracy and testing accuracy.

Lakhwani et al. (2020) use PIDD and a three-layered Artificial Neural Network (ANN) for DM diagnose [9]. Quasi Newton is used for training process. This research shows ANN is appropriate for DM prediction.

Patra and Khuntia (2021) utilise standard deviation K nearest neighbour (SDKNN) algorithm to classify PIDD so as to predict DM accurately [4]. This method consisits of three steps. The basic KNN is firstly applied for classification, and secondly the KNN with standard deviation (SD) is utilised, and finally the SD is used to find the nearest neighbour of each attribute so as to improve the accuracy. 90% of the dataset are used for training and 10% of them are used for training. The classification accuracy of this method achieves 83.2%.

Hasan et al. (2020) also uses PIDD to investigate DM diagnoses. This research measures different machine learning (ML) models with Area Under ROC Curve (AUC) metrics [6]. The dataset is firstly preprocessed by replacing missing values with mean values and removing outliers. The proposed model is a combination of enabling classifiers such as KNN, decision tree (DT), random forest (RF), Naive Bayes (NB), AdaBoost (AB), XGBoost (XB) and multilayer perception (MLP). This model achieves 0.789 for sensitivity, 0.934 for specificity, 0.092 for false omission rate, 66.234 for diagnostic odds ratio, and 0.950 for AUC.

Vaishali et al. (2017) employ genetic algorithm in PIDD to reduce the attributes of the dataset [10]. The input features is reduced from eight to four including age, BMI, glucose and diabetes pedigree function. 70% of the dataset are used for training and 30% of the dataset are used for testing. This research utilise MOE fuzzy algorithm to measure the performance of feature deduction. The accuracy achieves 83.0435% after feature reduction while the accuracy is 78.2609% without feature reduction.

Hayashi and Yukita (2016) choose Recursive-Rule eXtraction (Re-RX) with J48graft method to make classifications for PIDD [11]. The accuracy achieves 83.83% after using 10-fold cross validation for ten times. The Re-Rx method is chosen because it is considered as white box model with clear classification.

R programming language is widely used in DM diagnoses based on ML models.

Chang et al. (2022) use R programming language to compare three ML algorithms (J48 DT, RF and NB) for DM diagnoses [12]. PIDD is used as the dataset. 538 instances are used for training and 230 instances are used for testing. Feature selection is also applied to preprocess the dataset. The result shows that NB performs better with binary features while RF is good with more attributes.

Reddy et al. (2021) use R programming language to implement supervised learning models such as RF and KNN in PIDD [13]. 80% of the data are used for training and 20% of the data are used for testing. The training accuracy of RF and KNN reaches 75.1% and 76.5% separately, and the testing accuracy of RF and KNN reaches 78.4% and 80.8% separately.

Aada and Tiwari (2019) suggest R tool written by R language as a good environment because it performs well in vsualisation and data preprocessing [14]. This research compare the performance of support vector machine (SVM), Adaboost, linear regression (LR) and DT classifiers in DM diagnoses. Adaboost and SVM receives the best accuracy of 94.44% after bootstrapping, while DT only reaches 74.89%.

Tigga and Garg (2020) use logistic regression, KNN, SVM, NB, DT and RF algorithms to investigate PIDD and their own dataset so as to study DM diagnoses [15]. RStudio and R programming language are chosen for implementation. This research argues that they have provide a better dataset than PIDD since every algorithm could reach a higher accuracy in their dataset than in PIDD. RF reaches the highest accuracy of 94.10% in their own dataset.

Kaur and Kumari (2020) choose R programming language to investigate DM diagnoses using PIDD [16]. Feature selection is done by Boruta wrapper algorithm. Five different algorithms including SVM, SVM-linear, KNN, ANN and multifactor dimensionality reduction (MDR) are evaluated by the metrics of accuracy, AUC, recall, F_1 score and precision. All of these models have received good performance. SVM-linear achieves the highest accuracy with 0.89 and the second highest AUC with 0.90. KNN achieves the highest AUC with 0.92 and the second accuracy with 0.88. Therefore, SVM-linear and KNN are considered as the best models for DM diagnoses.

3 Methodology

The methodology chapter of this research consists of five primary steps: exploratory data analysis (EDA), data preprocessing, modelling, testing and performance evaluation. We would follow these five steps for data mining. In practice, we will preprocess the data while we analyse it.

3.1 Exploratory Data Analysis

We would explore the dataset, including the types of dataset features, the values of the most, mean and variance of each feature, etc. We will base our pre-processing of the dataset on the results of these analyses.

3.2 Data preprocessing

Hayashi and Yukita (2016) argues the existence of mislabeled or irregular data in the PIDD dataset would influence performance of the algorithm [11]. Hasan et al. (2020) also states that the preprocessing is needed since the existence of missing values and outliers in the dataset [6]. Therefore, it is necessary and meaningful to pre-process the data. It would be very effective in improving the accuracy of our training and testing models. For comparison purposes, we will use both pre-processed and raw data.

The pre-processing of data includes data transformation, data cleaning, data splitting and data reduction, etc. Data transformation is a process of converting data in the dataset to the type required by the model or algorithm so that it can be used. Data cleaning is a procedure that deals with missing or invalid values in a data set. And the data splitting is obviously for the purpose of training models, testing models and resampling. In contrast to them, data reduction is a common and effective technique to use when working with high-dimensional and complex data. We may not always solve the data if the correlation of some features is too high. Popular methods of data reduction include: Principal component analysis (PCA), Kernel PCA (KPCA) and C-mean clustering.

Notice that only one resampling method, which is 10-Fold Cross-validation, is used in this paper. The preprocessing methods used in this paper are, sequentially, using the KNN method to replace implausible values, a simple Box-Cox transformation, log transformation, normalizing numeric data to have a standard deviation of one and a mean of zero.

3.3 Modelling

Eight models including three common classification methods, three tree-based methods, one SVM method and one deep learning method are used in this paper.

A. Logistic Regression

Logistic regression (LR) is used to deal with regression problems where the dependent variable is a categorical variable. The common problem is binary or binomial distribution, and it can also deal with multi-category problems. It is actually a classification method. The relationship between the probability and the independent variables of the binary classification problem is often an S-shaped curve, which is realized by the Sigmoid function. The logistic/Sigmoid function is shown as below [15]:

$$y = \frac{1}{1 + e^{-x}}$$

The definition domain of the function is all real numbers, and the value range is between [0, 1]. The result corresponding to the x-axis at 0 is 0.5. If the output is greater than 0.5, it can be regarded as a 1. Otherwise, the result could be regarded as 0. If the output is just 0.5, it can be regarded as either 0 or 1 [15].

B. Naïve Bayes

Naïve Bayes (NB) model is a probalilistic machine learning method based on Bayes' theorem. It is considered as a simple and good algorithm. The function for calculating the posterior probability is as follows [15]:

$$p(c|x) = \frac{p(x|c)p(c)}{p(x)}$$

In this Bayes' theorem function, P(x|c) = Likelihood, P(c) = Class Prior Probability, P(x) = Predictor Prior Probability and P(c|x) = Posterior Probability [15].

C. K-Nearest Neighbor

K-Nearest Neighbor (KNN) algorithm is used to solve problems related to regression and classification. The main advantages of this algorithm are its translation simplicity and time efficiency. In the example Figure 1, point (2.5, 7) and (5.5, 4.5) could be assigned to either cluster. Euclidean distance is calculated by KNN to find the distance between the existing data points and new data points. Therefore (2.5, 7) belongs to the green cluster while (5.5, 4.5) belongs to the red cluster [15].

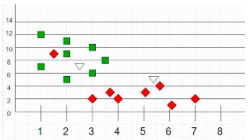


Figure 1. KNN Example [15]

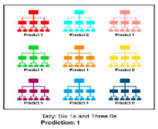


Figure 2. RF Example [15]

D. Random Forest

In machine learning, random forest (RF) is a classifier that consists of multiple decision trees, and the classes of its output are determined by the mode of the classes of the individual trees' outputs [15].

E. Bayesian Additive Regression Trees

Bayesian Additive Regression Trees (BART) is a decision tree model. The BART model has two main points. One is that the basic form of the model is the sum of multiple CART decision trees (sum-of-trees), and the other is that the model normalizes the prior of the parameters (regularization prior). The function of BART is shown as below [17]:

$$\hat{f}^b(x) = \sum_{k=1}^{K} \hat{f}_k^b(x), \text{ for } b = 1, 2, \ldots, B.$$

F. *Boosted Trees*

Boosted tree is also known as Gradient Boosting Decision Tree (GBDT), Gradient Boosting Regression Tree (GBRT) and Multiple Additive Regression Tree (MART) [18]. Boosting is a method for improving the prediction results of a decision tree. Boosting could be used for both regression and classification [17]. The function of he model is shown as:

$$f(x) = \sum_{k=1}^{K} f_k(x_i)$$

Where f_k means the space of functions containing all classification trees [18].

G. *Support Vector Machines (SVM)*

In machine learning, SVM is a supervised classifier mainly used for classification. SVM use hyperplane to classify points in a multidimensional space. A hyperplane is a N-1 dimensional child space defined in a N-dimensional space. In other words, a hyperplane is a line in a two dimensional space and a plane in three dimensional space. The hyperplane works as a boundary and classifies data points in order to create the maximum margin between the boundary and the classes [15].

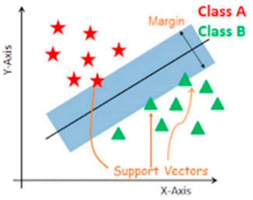
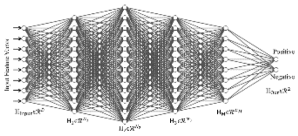


Figure 3. SVM Example [15]

H. *Multilayer Perceptron Model*

A multilayer perceptron (MLP) model is also known as a feed-forward neural network. Neurons are the processing units of a neural network and connected to each other according to different weights. A MLP model consists of an input layer, a output layer and multiple hidden layers. As shown in Figure 4, HM means the hidden layers and NM means the neurons of each hidden layer [6]. This research use the MLP model with a single layer as shown in Figure 5.



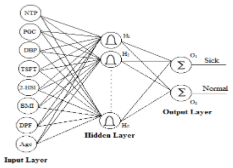


Figure 4. MLP Architecture with M hidden layers [6]   Figure 5. MLP with one hidden layer [2]

To implement the above eight models, this research used the tidy models and parsnip packages of R language. The model is trained with the segmented training set. Then the model is tested with the 10-fold cross validation method. Based on the results, the set of parameters with the best Area Under ROC Curve is selected and used in the final model.

3.4 *Testing*

Applying the previously split test set to our trained final model. Then the obtained prediction results and the real result data are used for model evaluation.

3.5 *Performance evaluation*

Accuracy, precision, recall, F-measures, ROC curve and AUC are the evaluation metrics in this paper. The following concepts are introduced so as to discuss the metrics [1]:

- True Positive(TP): Predicted to be positive, and actually positive
- True Negative(TN): Predicted to be negative, and actually negative
- False Positive(FP): Predicted to be positive, but actually negative
- False Negative(FN): Predicted to be negative, but actually positive

In a nutshell: true and false indicate whether you predicted correctly, whilst positive and negative indicate positive or negative predictions.

3.5.1 Accuracy

Accuracy is one of the most basic and widely used metrics. It represents the probability of correctly predicting, i.e. it equals to the number of correct predictions divided by the total number of predictions. It is calculated as:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

3.5.2 Precision

Precision (a.k.a. positive predictive value) indicates the percentage of correctly predicted positives to the total predicted positives, so false positive predictions would penalise your results. The range of this metric is between 0 and 1. It is calculated as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

3.5.3 Recall

Recall (a.k.a. sensitivity or true positive rate) shows what proportion of all positive cases are correctly classified as positive. The range of this metric is between 0 and 1. It is calculated as:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

3.5.4 F_1-Measures

F-Measure, a.k.a. F-Score, is a weighted harmonic mean of Precision and Recall. F-Measure is given by:

$$F_\beta = \frac{\beta^2 + 1}{\beta^2} \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

When the parameter b = 1, it becomes the most common F_1-Measure:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

3.5.5 ROC Curves and AUC

Receiver Operating Characteristic (ROC) Curve, also called sensitivity curve. It is named so because the points on the curve reflect the same sensitivity. They are all responses to the same signal stimulus, but they are the results of several different judgement criteria. AUC (Area Under Curve) is defined as the area under the ROC curve enclosed by the axes, which is obviously not greater than 1. Since the ROC curve is generally above the line y=x, the AUC can be taken to be between 0.5 and 1. The closer the AUC is to 1.0, the higher the authenticity of the test method. If it is equal to 0.5, the authenticity is the lowest and has no application value [17].

4 Results and Performances

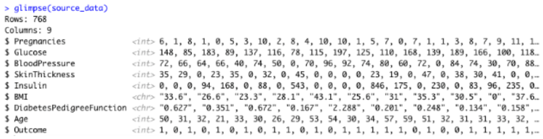
After loading packages and importing data, we started to analyse and preprocess the data.

4.1 Exploratory Data Analysis (EDA) and Data Preprocessing

As we all known, data analysis and data pre-processing are two steps having relevance. The reason why we analyse the data is for us to pre-process the data in a better way. This leads to improved results in the subsequent classification step. So we have merged these two steps together. When we have analysed some issues, we do some processing to the data so that we can do further analysis.

4.1.1 Data Transformation

At the very first step, let us take a glimpse at the dataset, and check the data types of the individual features within the dataset.



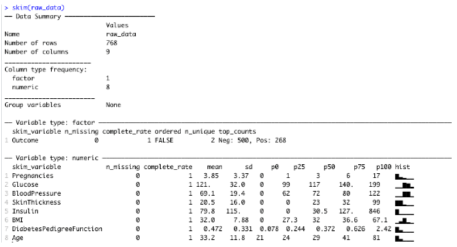
```
> glimpse(source_data)
Rows: 768
Columns: 9
$ Pregnancies              <int> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, 5, 7, 0, 7, 1, 1, 3, 8, 7, 9, 11, 1…
$ Glucose                  <int> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125, 110, 168, 139, 189, 166, 100, 118…
$ BloodPressure            <int> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 74, 80, 60, 72, 0, 84, 74, 30, 70, 88…
$ SkinThickness            <int> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0, 23, 19, 0, 47, 0, 38, 30, 41, 0, 0,…
$ Insulin                  <int> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0, 846, 175, 0, 230, 0, 83, 96, 235, 0…
$ BMI                      <chr> "33.6", "26.6", "23.3", "28.1", "43.1", "25.6", "31", "35.3", "30.5", "0", "37.6…
$ DiabetesPedigreeFunction <chr> "0.627", "0.351", "0.672", "0.167", "2.288", "0.201", "0.248", "0.134", "0.158",…
$ Age                      <int> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 34, 57, 59, 51, 32, 31, 31, 33, 32, …
$ Outcome                  <int> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,…
```

Figure 6. Data types for each feature

Both the BMI & DiabetesPedigreeFunction columns are non-numeric, need to be converted. Meanwhile, the Outcome column need to be transform into no-numeric in order to make it easier to categorize later.

4.1.2 Data Cleaning

After completing the data transformation, take a look at the abbreviated information on the data set, including means, variances and distribution histograms., etc.



```
> skim(raw_data)
── Data Summary ────────────────────────
                           Values
Name                       raw_data
Number of rows             768
Number of columns          9
_______________________
Column type frequency:
  factor                   1
  numeric                  8
________________________
Group variables            None

── Variable type: factor ──────────────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate ordered n_unique top_counts
1 Outcome               0             1 FALSE          2 Neg: 500, Pos: 268

── Variable type: numeric ─────────────────────────────────────────────────────────────────────
  skim_variable            n_missing complete_rate    mean     sd    p0    p25    p50    p75  p100 hist
1 Pregnancies                      0             1   3.85   3.37  0        1      3      6     17  ▇▃▂▁▁
2 Glucose                          0             1 121.    32.0   0       99    117    140.   199  ▁▁▇▆▂
3 BloodPressure                    0             1  69.1   19.4   0       62     72     80    122  ▁▁▇▇▁
4 SkinThickness                    0             1  20.5   16.0   0        0     23     32     99  ▇▇▂▁▁
5 Insulin                          0             1  79.8  115.    0        0     30.5  127.   846  ▇▁▁▁▁
6 BMI                              0             1  32.0    7.88  0       27.3   32     36.6   67.1 ▁▃▇▂▁
7 DiabetesPedigreeFunction         0             1   0.472  0.331 0.078  0.244  0.372  0.626  2.42 ▇▃▁▁▁
8 Age                              0             1  33.2   11.8  21       24     29     41     81  ▇▃▁▁▁
```

Figure 7. A skim of raw data

We found that there is no missing value. But we notice that the minimum value (p0) for the Glucose, BloodPressure, Skin Thickness, Insulin and BMI columns is 0. But these do not make sense. Because as long as it is a alive person, these data will not be zero. Therefore we have to process them.

Through counting the zero values in these columns, we also found that there were 374 and 227 zero values for Insulin and SkinThickness, which were among the five attributes we mentioned above, respectively. Therefore, if we remove the rows containing zero values in all five features, our dataset size will be almost halved.

After our discussion, we decided to replace the corresponding zero values by using the weighted average of the five nearest neighbors. However, this processing will follow the data splitting. Meanwhile, we also keep the unprocessed data. We train the model with both processed and unprocessed data and compare their differences.

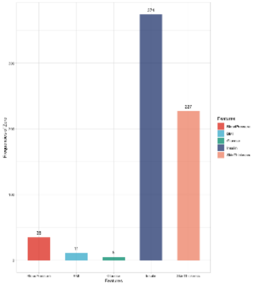


Figure 8. Zero frequencies of five specific features

4.1.3 Data Reduction

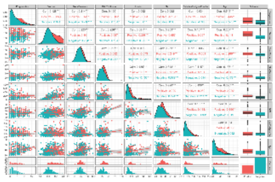


Figure 9. A matrix of plots with raw data

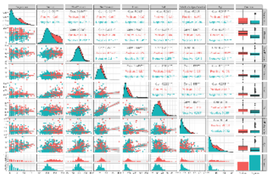


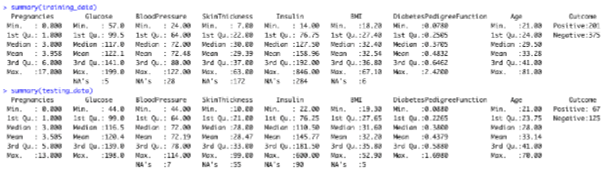
Figure 10. A matrix of plots with processed data

The three pairs of variables with the highest correlations were SkinThickness and BMI, Insulin and Glucose, and Age and Pregnancies, with correlations of 0.659, 0.606 and 0.544 respectively, in processed dataset.

In contrast, the correlation coefficients for these three groups of variables were also the highest in the raw dataset, at 0.648, 0.581 and 0.544 respectively. Apart from this, no other correlation coefficient exceeds 0.5. Thus we did not see the need to reduce the dimensionality of the data.

4.1.4 Data Splitting

We divided the dataset into a training set and a test set, with 75% of the training set and 25% of the test set. The training set is then divided into 10 folds for cross-validation, and finally the best model parameters are selected.



```
> summary(training_data)
  Pregnancies        Glucose       BloodPressure     SkinThickness       Insulin            BMI        DiabetesPedigreeFunction       Age              Outcome
 Min.   : 0.000   Min.   : 57.0   Min.   : 24.00   Min.   : 7.00   Min.   : 14.00   Min.   :18.20   Min.   :0.0780            Min.   :21.00   Positive:201
 1st Qu.: 1.000   1st Qu.: 99.5   1st Qu.: 64.00   1st Qu.:22.00   1st Qu.: 76.75   1st Qu.:27.40   1st Qu.:0.2505            1st Qu.:24.00   Negative:375
 Median : 3.000   Median :117.0   Median : 72.00   Median :30.00   Median :127.50   Median :32.40   Median :0.3705            Median :29.50
 Mean   : 3.958   Mean   :122.1   Mean   : 72.48   Mean   :29.39   Mean   :158.96   Mean   :32.54   Mean   :0.4832            Mean   :33.28
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:37.00   3rd Qu.:192.00   3rd Qu.:36.80   3rd Qu.:0.6462            3rd Qu.:41.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :63.00   Max.   :846.00   Max.   :67.10   Max.   :2.4200            Max.   :81.00
                  NA's   :5       NA's   :28       NA's   :172     NA's   :284      NA's   :6
> summary(testing_data)
  Pregnancies        Glucose       BloodPressure     SkinThickness       Insulin            BMI        DiabetesPedigreeFunction       Age              Outcome
 Min.   : 0.000   Min.   : 44.0   Min.   : 44.00   Min.   :10.00   Min.   : 22.00   Min.   :19.30   Min.   :0.0880            Min.   :21.00   Positive: 67
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:21.00   1st Qu.: 76.25   1st Qu.:27.65   1st Qu.:0.2265            1st Qu.:23.75   Negative:125
 Median : 3.000   Median :116.5   Median : 72.00   Median :28.00   Median :110.50   Median :31.60   Median :0.3800            Median :28.00
 Mean   : 3.505   Mean   :120.4   Mean   : 72.19   Mean   :28.47   Mean   :145.77   Mean   :32.20   Mean   :0.4379            Mean   :33.14
 3rd Qu.: 5.000   3rd Qu.:139.0   3rd Qu.: 78.00   3rd Qu.:33.00   3rd Qu.:181.50   3rd Qu.:35.80   3rd Qu.:0.5880            3rd Qu.:41.00
 Max.   :13.000   Max.   :198.0   Max.   :114.00   Max.   :99.00   Max.   :600.00   Max.   :52.90   Max.   :1.6980            Max.   :70.00
                                  NA's   :7        NA's   :55      NA's   :90       NA's   :5
```

Figure 11. Summaries of training data and testing data

4.2 Classification

In this paper, four models, K-Nearest Neighbors, Random Forests, Bayesian additive regression trees and Support Vector Machines, only used processed data. And another four models, Logistic Regression, Naïve Bayes, Boosted Trees and Multilayer Perceptron Model used both raw data and processed data.

4.2.1 Training Result

The following are the results of the training.

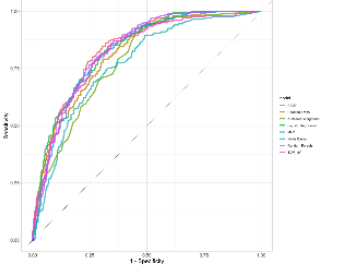
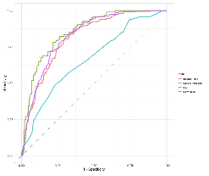


Figure 12. ROC Curves of Validation Set for top penalized Models using Processed Data



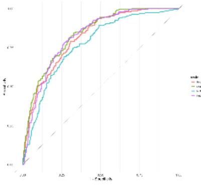
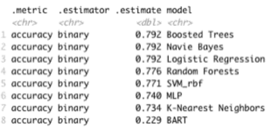


Figure 13. Four ROC Curves of Validation Set for top penalized Models using Raw Data

Figure 14. Four ROC Curves of Validation Set for top penalized Models using Processed Data

4.2.2 Evaluation

A. Accuracy



```
  .metric  .estimator .estimate model
  <chr>    <chr>          <dbl> <chr>
1 accuracy binary         0.792 Boosted Trees
2 accuracy binary         0.792 Navie Bayes
3 accuracy binary         0.792 Logistic Regression
4 accuracy binary         0.776 Random Forests
5 accuracy binary         0.771 SVM_rbf
6 accuracy binary         0.740 MLP
7 accuracy binary         0.734 K-Nearest Neighbors
8 accuracy binary         0.229 BART
```

Figure 15. Accuracy ranking of the eight models in the case of using processed data

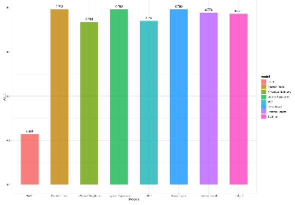
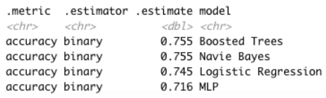


Figure 16. An overview of the accuracy for eight models using the processed data



```
  .metric  .estimator .estimate model
  <chr>    <chr>          <dbl> <chr>
1 accuracy binary         0.755 Boosted Trees
2 accuracy binary         0.755 Navie Bayes
3 accuracy binary         0.745 Logistic Regression
4 accuracy binary         0.716 MLP
```

Figure 17. Accuracy ranking of the four models in the case of using raw data

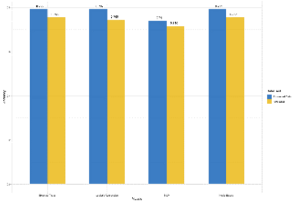
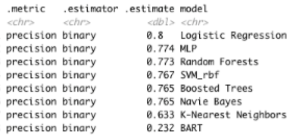


Figure 18. Comparison of the accuracy results for four models using different training sets

B. Precision



```
  .metric   .estimator .estimate model
  <chr>     <chr>          <dbl> <chr>
1 precision binary         0.8   Logistic Regression
2 precision binary         0.774 MLP
3 precision binary         0.773 Random Forests
4 precision binary         0.767 SVM_rbf
5 precision binary         0.765 Boosted Trees
6 precision binary         0.765 Navie Bayes
7 precision binary         0.633 K-Nearest Neighbors
8 precision binary         0.232 BART
```

Figure 19. Precision ranking of the four models in the case of using processed data

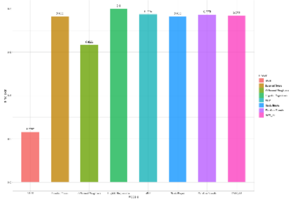
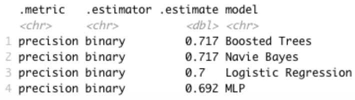


Figure 20. An overview of the precision for eight models using the processed data



```
  .metric   .estimator .estimate model
  <chr>     <chr>          <dbl> <chr>
1 precision binary         0.717 Boosted Trees
2 precision binary         0.717 Navie Bayes
3 precision binary         0.7   Logistic Regression
4 precision binary         0.692 MLP
```

Figure 21. Precision ranking of the four models in the case of using raw data

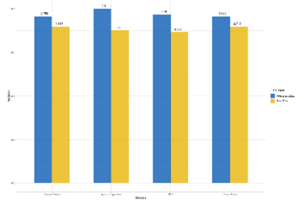
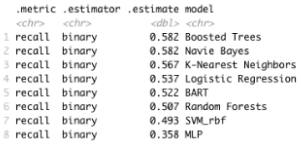


Figure 22. Comparison of the precision results for four models using different training sets

C. Recall



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 recall  binary         0.582 Boosted Trees
2 recall  binary         0.582 Navie Bayes
3 recall  binary         0.567 K-Nearest Neighbors
4 recall  binary         0.537 Logistic Regression
5 recall  binary         0.522 BART
6 recall  binary         0.507 Random Forests
7 recall  binary         0.493 SVM_rbf
8 recall  binary         0.358 MLP
```

Figure 23. Recall ranking of the four models in the case of using processed data

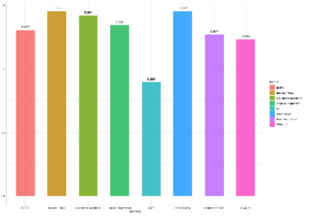
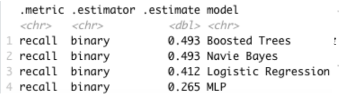


Figure 24. An overview of the recall for eight models using the processed data



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 recall  binary         0.493 Boosted Trees
2 recall  binary         0.493 Navie Bayes
3 recall  binary         0.412 Logistic Regression
4 recall  binary         0.265 MLP
```

Figure 25. Recall ranking of the four models in the case of using raw data

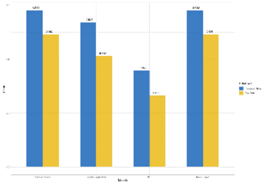
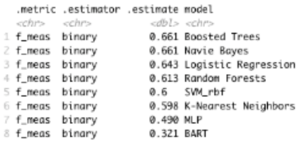


Figure 26. Comparison of the recall results for four models using different training sets

D. F1-measure



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 f_meas  binary         0.661 Boosted Trees
2 f_meas  binary         0.661 Navie Bayes
3 f_meas  binary         0.643 Logistic Regression
4 f_meas  binary         0.613 Random Forests
5 f_meas  binary         0.6   SVM_rbf
6 f_meas  binary         0.598 K-Nearest Neighbors
7 f_meas  binary         0.490 MLP
8 f_meas  binary         0.321 BART
```

Figure 27. F1-measure ranking of the four models in the case of using processed data

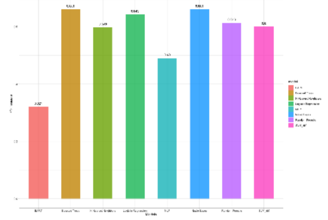
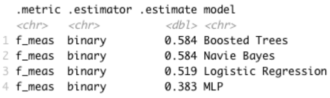


Figure 28. An overview of the F1-measure for eight models using the processed data



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 f_meas  binary         0.584 Boosted Trees
2 f_meas  binary         0.584 Navie Bayes
3 f_meas  binary         0.519 Logistic Regression
4 f_meas  binary         0.383 MLP
```

Figure 29. F1-measure ranking of the four models in the case of using raw data

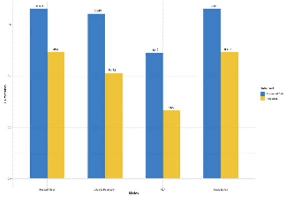


Figure 30. Comparison of the F1-measure results for four models using different training sets

E. ROC Curves

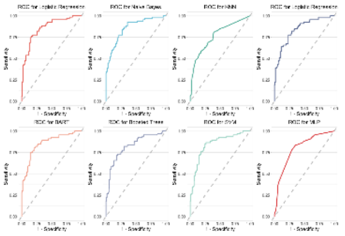


Figure 31. ROC curves of eight models using the processed data

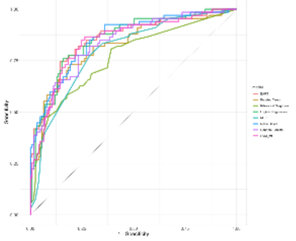


Figure 32. ROC curves for the eight models using the processed data in one coordinate system

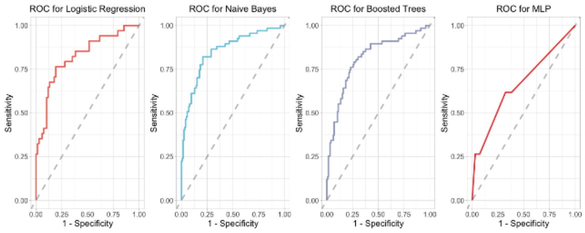


Figure 33. ROC curves of four models using the raw data

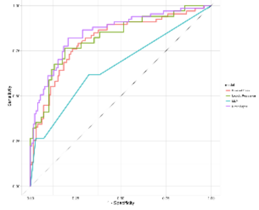


Figure 34. ROC curves for the four models using the raw data in one coordinate system

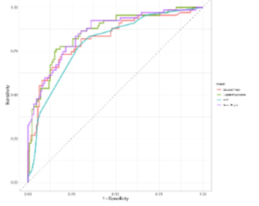
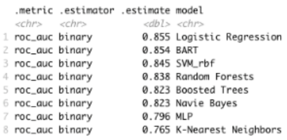


Figure 35. ROC curves for the four models using the processed data in one coordinate system

F. Area Under Curve (AUC)



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 roc_auc binary         0.855 Logistic Regression
2 roc_auc binary         0.854 BART
3 roc_auc binary         0.845 SVM_rbf
4 roc_auc binary         0.838 Random Forests
5 roc_auc binary         0.823 Boosted Trees
6 roc_auc binary         0.823 Navie Bayes
7 roc_auc binary         0.796 MLP
8 roc_auc binary         0.765 K-Nearest Neighbors
```

Figure 36. AUC ranking of the eight models in the case of using processed data

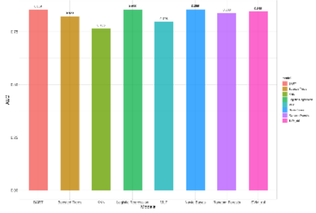
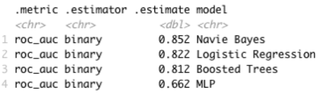


Figure 37. An overview of the AUC for eight models using the processed data



```
  .metric .estimator .estimate model
  <chr>   <chr>          <dbl> <chr>
1 roc_auc binary         0.852 Navie Bayes
2 roc_auc binary         0.822 Logistic Regression
3 roc_auc binary         0.812 Boosted Trees
4 roc_auc binary         0.662 MLP
```

Figure 38. AUC ranking of the four models in the case of using raw data

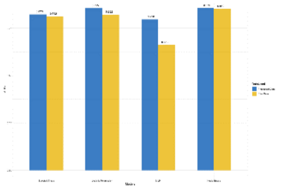


Figure 39. Comparison of the AUC results for four models using different training sets

G. Variables Importance

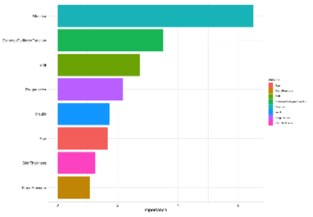


Figure 40. Variables importance for logistic regression using the processed data

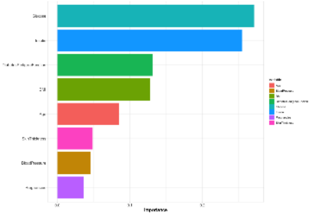


Figure 41. Variables importance for boosted trees using the processed data

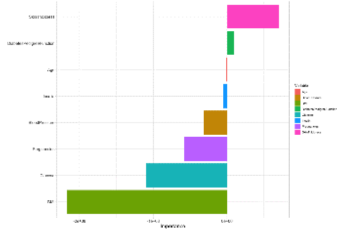


Figure 42. Variables importance for MLP using the processed data

5 DISCUSSION

Figure 12 shows the best ROC curves for each model trained using the processed data. It can be readily seen that the majority of models have AUC values between 0.75 and 0.85, which is an acceptable result.

Figures 13 and 14 demonstrate the best results obtained from four models, Boosted Trees, Logistic Regression, MLP and Navies Bayes, trained using processed and unprocessed data. respectively. It is clear that using processed data to train the model can significantly improve the training results. Using processed data gives a smoother ROC curve than using unprocessed data.

Through Figures 15 and 16, we can learn that with the models trained using the processed data, five of the eight exceeded 0.75 in accuracy, seven exceeded 0.7, and even three all achieved 0.792. The BART model performed the worst with an accuracy of only 0.229, which is an unacceptable outcome. Removing a maximum and a minimum accuracy, their average accuracy reaches 0.7675. Comparing the accuracy results obtained by the four models using different data for training respectively. It is evident that the processing techniques used in this paper are effective in improving the accuracy. The average improvement in accuracy for the four models reached 3.6%.

Figures 19 and 20 show the precision results obtained for the eight models trained using the processed data. Similar to the accuracy results, the BART model performs the worst with a precision of merely 0.232, followed by the KNN model with a precision of 0.633. The rest of the models have precision in the range of 0.75 to 0.8.

As shown in Figure 21 and Figure 22, the data pre-processing method used in this paper achieves an average improvement in precision of 6.95%. This far exceeds the improvement to accuracy. This result is quite satisfactory. Figures 23 and 24 demonstrated recall results that were not as impressive as the former accuracy and precision. The overall recall ranged from 0.5 to 0.6. The BART model achieved an average recall, but the MLP model had a slightly surprising recall of only 0.358.

Analysing Figures 25 and 26 it is easy to notice that the processed training data resulted in an average improvement in recall values reaching 10%.

F1-measure is twice the harmonic mean of precision and recall. Under this metric, the unstable BART and MLP models had only 0.321 and 0.49, respectively. The remaining F1-measure was distributed between 0.6 and 0.66.

The enhancement effect of processing the training set on the F1-measure was the most pronounced among the previous four metrics. The boost in F1-measure reached a remarkably impressive averaging of 19.8%.

Figure 31 illustrates the final predicted ROC curves for each of the eight models. The eight models in this figure were all trained using the processed data. For a clearer comparison of their performance, they have been placed in a single coordinate system in Figure 32.

Their final predicted ROC curves for the four models trained with unprocessed data are shown in Figure 33. In Figure 34, they are presented in the same coordinate system. Correspondingly, Figure 35 displays the ROC curves for the final predictions of these four models trained using the processed data. It is evident that training with the processed data in general improves the final predicted ROC curve slightly. Also, the ROC curve of the MLP model was unexpectedly improved dramatically.

According to Figure 36 and Figure 37, the final predicted AUCs obtained for all eight models trained with the processed data achieved reasonably satisfactory results. The AUC of all models exceeded 0.76. Except for MLP and KNN models, the AUC of all models exceeded 0.8. The logistic regression model yielded the highest AUC with 0.855. It is a remarkably impressive result.

Considering Figure 38 and Figure 39, training with the processed data brought a noticeably lower boost to AUC than to Precision, Recall and F1-Measures. Apart from the 13% improvement in AUC for the MLP model, the improvement for the other three models was quite minor, at around a 1% level.

By analysing Figures 40 and 41 , it becomes clear that the most important variable influencing both the logistic regression and boosted tree models is Glucose, followed by the three variables Diabetes Pedigree Function, BMI and

Pregnancies. However, Figure 42 reflects a confusing result, with only Skin Thickness and Diabetes Pedigree Function providing positive importance and the rest of the variables all offering negative importance.

It is obvious that the results acquired in this paper, no matter the Accuracy, Precision, Recall, F1--Measures or ROC and AUC, are hardly on the same level as the results in literature reviews. This is mainly attributed to two reasons. One reason is that our pre-processing of the source data is still too simple. The methods used in the literature to pre-process the data are generally more complex than those used in this paper. Another extremely important reason is that the models used in this paper are derived from the parsnip package. In order to make the comparison of the eight models meaningful, we did not tune the parameters and hyperparameters of any model, but used the default settings of the most basic models. Therefore, the results in this paper are considerably less favourable than those in the literature.

6 Conclusion and Future Research

It can be found that among the eight models, Logistic Regression gave the best performance in terms of accuracy and AUC metrics, with 0.792 and 0.855 respectively. Closely followed by four models, Random Forest, SVM, Naïve Bayes and Boosted Trees, with accuracies and AUCs above 0.77 and 0.82. The least stable is BART, which has the worst Accuracy at 0.229 and the third highest AUC at 0.845. Perhaps tuning the parameters of the BART model would ameliorate its instability. Excluding the BART model, the KNN model, which performed the worst, had Accuracy of 0.734 and AUC of 0.765.

Apart from the non-stationary BART model, the remaining seven models, after training with PIDD, predicted the diagnosis of DM with accuracies above 0.73 and an average accuracy of 0.771, which is a promising result. One could suggest that these seven models are highly effective in the diagnosis of DM.

Based on the discussion section, future research could be focused on the following two areas. One is using more sophisticated and efficient data pre-processing methods. Another is to tune the parameters and hyperparameters of the models for more accurate prediction results. Furthermore, we can also use methods not employed in this paper, such as C5.0 rule-based classification models, Decision trees, Linear/Quadratic discriminant analysis, etc., to do our research.

Contribution

This research is conducted by Rui Wang (20396662) and Yue Xue (20406680). Rui Wang implemented four models, Logistic Regression, Naïve Bayes, Boosted Trees and MLP, using both raw data and processed data to investigate DM diagnoses. Yue Xue implemented four models, K-Nearest Neighbors, Random Forest, Bayesian additive regression trees (BART) and Radial basis function Support Vector Machines (SVM), using only processed data to investigate DM.

References

[1] Sivanesan, R. and Dhivya, K. D. R. (2017) A Review on diabetes mellitus diagnoses using classification on Pima Indian diabetes data set. **International Journal of Advance Research in Computer Science and Management Studies** 5(1): pp. 12-17.

[2] Soltani, Z. and Jafarian, A. (2016) A new artificial neural networks approach for diagnosing diabetes disease type II. **International Journal of Advanced Computer Science and Applications** 7(6): pp. 89-94.

[3] World Health Organization (2019) **Classification of diabetes mellitus** [online]. Available at: https://www.who.int/publications/i/item/classification-of-diabetes-mellitus [Accessed 30 April 2022].

[4] Patra, R. and Khuntia, B. (2021) Analysis and prediction of Pima Indian Diabetes Dataset using SDKNN classifier technique. **IOP Conference Series: Materials Science and Engineering** 1070(1): pp. 012059.

[5] American Diabetes Association Professional Practice Committee. (2022) 2. Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes-2022. **Diabetes Care** 45(Supplement 1): pp.S17-S38.

[6] Hasan, M.K., Alam, A., Das, D., Hossain, E. and Hasan, M. (2020) Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers. **IEEE Access** 8: pp.76516–76531.

[7] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C. and Johannes, R.S. (1988) Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. **In Proceedings of the Symposium on Computer Applications and Medical Care, American Medical Informatics Association.** pp. 261-265. Paris: IEEE Computer Society Press.

[8] Kaggle (2016) **Pima Indians Diabetes Database** [online]. Available at: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database [Accessed 30 April 2022].

[9] Lakhwani, K., Bhargava, S., Hiran, K. K., Bundele, M. M. and Somwanshi, D. (2020) Prediction of the onset of diabetes using artificial neural network and pima indians diabetes dataset. **5th IEEEInternational Conference on Recent Advances and Innovations in Engineering (ICRAIE)** pp. 1-6.

[10] Vaishali, R., Sasikala, R., Ramasubbareddy, S., Remya, S. and Nalluri S. (2017) Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset. **International Conference on Computing Networking and Informatics (ICCNI)** pp. 1-5.

[11] Hayashi, Y. and Yukita, S. (2016) Rule extraction using Recursive-Rule extraction algorithm with J48graft combined with sampling selection techniques for the diagnosis of type 2 diabetes mellitus in the Pima Indian dataset. **Informatics in Medicine Unlocked** 2: pp. 92–104.

[12] Chang, V., Bailey, J., Xu, Q. A. and Sun, Z. (2022) Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms. **Neural Computing and Applications** pp.1-17.

[13] Reddy, S. K., Krishnaveni, T., Nikitha, G. and Vijaykanth, E. (2021) Diabetes Prediction Using Different Machine Learning Algorithms. **2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)** pp. 1261-1265.

[14] Aada, A. and Tiwari, S. (2019) Predicting diabetes in medical datasets using machine learning techniques. **International Journal of Scientific Research and Engineering Trends** 5: pp.257-267.

[15] Tigga, N.P. and Garg, S. (2020) Prediction of Type 2 Diabetes using Machine Learning Classification Methods. **Procedia Computer Science** 167: pp. 706–716.

[16] Kaur, H. and Kumari, V. (2020) Predictive modelling and analytics for diabetes using a machine learning approach. **Applied Computing and Informatics** 18(1/2): pp. 90-100.

[17] Gareth, J., Daniela, W., Trevor, H. and Robert, T. (2013) **An introduction to statistical learning: with applications in R.** New York: Springer.

[18] Chen, T. (2014) Introduction to boosted trees. **University of Washington Computer Science** 22(115): pp.14-40.