



ON COMPARING MULTI-LAYER PERCEPTRON AND LOGISTIC REGRESSION FOR CLASSIFICATION OF DIABETIC PATIENTS IN FEDERAL MEDICAL CENTER YOLA, ADAMAWA STATE



H. Ahmed¹  , M. B. Mohammed² and I. A. Baba³

¹Department of Statistics and Operations Research, Modibbo Adama University of Technology, Yola, Nigeria

²Department of Mathematics and Computer Science, Federal University of Kashere, Gombe, Nigeria

³Department of Mathematical Sciences, Taraba State University, Jalingo, Nigeria



ABSTRACT

The logistic regression (LR) and Multi-Layer Perceptron (MLP) are used to handle regression analysis when the dependent response variable is categorical. Therefore, this study assesses the performance of LR and MLP in terms of classification of object/observations into identified component/groups. A data set consists of 553 cases of diabetes mellitus were collected at Federal Medical Center, Yola. The variables measured: Age(years), Mass of a patient(kg/meters), glucose level (plasma glucose concentration, a 2-hour in an oral glucose tolerance test), pressure (Diastolic blood pressure mmHg), insulin (2-hour serum insulin μ U/ml) and class variable (0 or 1) treating 0 as false or negative and 1 treated as true or positive test for diabetes. The method used in the study is Logistic regression analysis and the multi-Layer perceptron, a type of Artificial Neural Network, confusion matrix, classification, network algorithm and SPSS version 21 for Windows 10.1. The result of the study showed that LP classifies diabetic patients correctly with 91.8% accuracy. While it classifies non-diabetic patients with 89.1% accuracy. MLP classifies diabetic patients with 88.6% accuracy while it classifies non-diabetic patients with 93.2% classification accuracy. Overall, MLP classifies better with 91% accuracy while LR classifies with 90.6% accuracy. This study complements other literatures where MLP, a type Artificial neural network classifies and predicts better than other non-neural network classifiers.

Received 16 May 2021

Accepted 31 May 2021

Published 25 June 2021

Corresponding Author

H. Ahmed, hassanahmed.official@gmail.com

DOI [10.29121/ijetmr.v8.i6.2021.961](https://doi.org/10.29121/ijetmr.v8.i6.2021.961)

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Copyright: © 2021 The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Keywords: Logistic Regression, MultiLayer Perceptron, Artificial Neural Network, Log Likelihood Ratio, Diabetes Mellitus

1. INTRODUCTION

Many statistical techniques are available for handling various problems. Some of these techniques come as models such as linear, exponential and quadratic models.



These models have become integral components concerned with describing the relationship between a response variable and one or more explanatory variables [D. Hosmer and Lemeshow \(2000\)](#). If there is a reason to believe that a linear relationship exists between a variable of interest (response variable) and other variables (predictor variables) in a study, the ordinary linear model is one technique that is often used for predicting outcomes [Alan \(2002\)](#). This technique is mostly adopted due to its flexibility for analyzing the relationship between multiple independent variables and a single dependent variable. Much of its flexibility is due to the way in which all sorts of independent variables can be accommodated [Joaquim and Sá \(2007\)](#).

It is meaningful to address how the analyst can deal with data representing multiple independent variables and a categorical dependent variable, how independent variables can be used to contribute to the discovery of differences in the categories. The assignment of observations or objects into predefined homogenous groups is a problem of major practical and research interest. For example, we may use quantitative information in predicting who will or will not graduate from a college. This would be an example of simple binary classification problems, where the categorical dependent variable can only assume two distinct values. In other cases, there are multiple categories or classes for the categorical dependent variable. For example, when we are ill, we want a doctor to diagnose our disease from the symptoms of the illness, the outcome may be more than two.

All the above are classification problems where we attempt to predict values of a categorical dependent variable from one or more continuous and/or categorical predictor variables. In statistics, it is the process of allocating an observation p in one of several predefined groups or categories and an ideal classification method which distinguishes different classes from each other. The basic objective is to build a discriminant function that takes the information to summarize the p variables on an indicator that yields the optimal discrimination between the classes – the goal of classification in this case – also known as supervised pattern recognition [Wehrens \(2010\)](#). In order to derive the decision rule that yields the optimal discrimination between the classes, one assumes that a training set of pre-classified cases – the data sample – is available, and can be used to determine the model applicable to new cases. The decision rule can be derived in a model-based approach, whenever a joint distribution of the random variables can be assumed, or in a model-free approach [Joaquim and Sá \(2007\)](#).

There are numerous algorithms for predicting continuous or categorical variables from a set of continuous predictors and/or categorical factor effects [Lewicki and T \(2006\)](#). For example, in GLM (General Linear Models) and GRM (General Regression Models), we can specify a linear combination design of continuous predictors and categorical factor effects to predict a continuous dependent variable. In GDA (General Discriminant Function Analysis), we can specify such designs for predicting categorical variables to solve classification problems.

A neural-network is a classification algorithm in the field of artificial intelligence. It is a very powerful tool with the capability of pattern recognition. Artificial Neural Networks (ANNs) were designed to model the functioning of human brain. Linear classifiers separate objects by the value of a linear combination of their features. The feature of an object is represented by a vector. There is another vector to be trained with known observations. This is called weight vector. There are several algorithms in this category such as Support Vector Machines (SVM), Multi-layer Perceptron (MLP) and the radial Basis Function (RBF).

The objective of this work is to evaluate the implementation and performance of classification techniques (a multi-layer perceptron) comparatively with a logistic regression model, in order to predict the presence of diabetes in a collected data from Federal Medical Center, Yola, Adamawa State, Nigeria. This paper describes how these techniques have been applied to the data and presents a comparison analysis. The results are reported and discussed according to this technique.

2. MATERIALS AND METHODS

Between 1st August, 2016 to 31st October, 2017, a total of five hundred and fifty-three (553) women were tested for diabetes at FMC, Yola. Three hundred and six were diabetic while two hundred and forty-seven were non diabetic. The data collected was from records of patients at FMC, YOLA.

Observation with missing data were dropped from the analysis. The final dataset consists of 553 subjects, described by several clinical characteristics.

The classification task consists of predicting whether a patient would test positive for diabetes. The class labels of the data are 1 for diabetes and 0 otherwise. There are 8 predictor variables for 553 patients.

The data set have the following numeric attributes and they are:

1. "glucose": Plasma glucose concentration 2 hours in an oral glucose tolerance test.
2. "pressure": Diastolic blood pressure (mm Hg)
3. "insulin": 2-Hour serum insulin (μ U/ml).
4. "mass": Body mass index (weight in kg/(height in meters)²)
5. "age": Age in years.
6. Class variable (0 or 1). The Class variable (6) is treated as 0 (false), 1 (true – tested positive for diabetes).

APPLICATION OF CLASSIFICATION TECHNIQUES.

Two classification techniques were used to fit a prediction model to the data; logistic regression and a multi-layer perceptron. This fitting process is hereby described for each method.

Logistic Regression model

Consider a random variable W that can take either of the two possible values. Given a dataset with a total sample size of M , where each observation is independent, W can be assumed as a Bernoulli random variable. On the prevalence of diabetes mellitus in the North Western Part of Nigeria 4 column vector of M binomial random variables W_i . By convention, a value of 1 is used to designate "success" and a value of 0 used to signify "failure." To simplify computational details of estimation, it is convenient to aggregate the data such that each row represents one distinct combination of values of the independent variables. These rows are often referred to as "populations." Let N represent the total number of populations and let n be a column vector with elements n_i representing the number of observations in population i for $i = 1$ to N and M , the total sample size. Now, let Y be a column vector of length N where each element Y_i is a random variable representing the number of successes of W for the population. Let column vector y contain elements y_i representing the observed counts of the number of successes for each population. Let π be a column vector also of length N with elements $\pi_i = p(w_i = 1)$, i.e., the probability of success for any given observation in the i th population. The linear components of the model include design matrix and the vector of parameters to be projected. The design matrix of independent variables, X , is composed of N rows and $K + 1$ columns, where K is the number of explanatory variables specified in the model, for each row of the design matrix, the first element $x_{i0} = 1$. This is the intercept or "alpha". The parameter vector, is a column vector of length $K + 1$. There is one parameter equivalent to each

of the K columns of independent variable settings in X , plus one β_0 , for the intercept. The logistic regression model equates the logit transforms, the log-odds of probability of a success, to the linear component:

$$\log \left(\frac{\pi_i}{1-\pi_i} \right) = \sum_{k=0}^K x_{ik} \beta_k \quad i = 1, 2, \dots, N. \quad (1)$$

Where $\left(\frac{\pi_i}{1-\pi_i} \right)$ is known as the odds of an event. Suppose y takes the values 1 for an event and 0 for a non-event, hence y has a Bernoulli distribution with probability parameter (and expected value) p .

Parameter estimation

The goal of logistic regression is to estimate the $K + 1$ unknown parameters β in (1). This is done with maximum likelihood estimation, which entails finding the set of parameters for which the probability of the observed data is greatest. The maximum likelihood equation is derived from the probability distribution of the dependent variable. Since each y_i represents a binomial count in the i th population, the joint probability density function of Y is:

$$f(y \setminus \beta) = \prod_{i=1}^N \frac{n_i!}{y_i! (n_i - y_i)!} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} \quad (2)$$

For each population, there are different ways to arrange y_i successes from n_i trials. Since the probability of success for any of n_i trials is π_i , the probability of y_i successes is $\pi_i^{y_i}$. Likewise, the probability of $n_i - y_i$ is $(1 - \pi_i)^{n_i - y_i}$. The joint probability densities function in (2) expresses the values of y as a function of known fixed values for β . Thus,

$$L(\beta \setminus y) = \prod_{i=1}^N \frac{n_i!}{y_i! (n_i - y_i)!} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} \quad (3)$$

The maximum likelihood estimates are the values for β that maximize the likelihood function in (3). Thus, finding the maximum likelihood estimates requires computing the first and second derivatives of the likelihood function. Attempting to take the derivative of (3) with respect to β , and after rearranging terms, the equation to be maximized can be written as:

$$\beta \setminus y) = \prod_{i=1}^N \left(\frac{\pi_i}{1 - \pi_i} \right)^{y_i} (1 - \pi_i)^{n_i} \quad (4)$$

Recall that:

$$\log \left(\frac{\pi_i}{1 - \pi_i} \right) = \sum_{k=0}^K x_{ik} \beta_k \quad (5)$$

After taking exponent on both sides, equation (5) becomes:

$$\left(\frac{\pi_i}{1-\pi_i}\right) = e^{\sum_{k=0}^K x_{ik}\beta_k} \quad (6)$$

After solving for π_i it becomes

$$\pi_i = \left(\frac{e^{\sum_{k=0}^K x_{ik}\beta_k}}{1+e^{\sum_{k=0}^K x_{ik}\beta_k}}\right) \quad (7)$$

Substituting (6) for the first term and (7) for the second term, equation (4) becomes:

$$\beta \setminus y = \prod_{i=1}^N (e^{\sum_{k=0}^K x_{ik}\beta_k})^{y_i} \left(1 - \frac{e^{\sum_{k=0}^K x_{ik}\beta_k}}{1+e^{\sum_{k=0}^K x_{ik}\beta_k}}\right)^{n_i} \quad (8)$$

Simplifying the product on the right-hand side in (8), it can be now be written as:

$$\beta \setminus y = \prod_{i=1}^N (e^{\sum_{k=0}^K x_{ik}\beta_k})^{y_i} (1 + e^{\sum_{k=0}^K x_{ik}\beta_k})^{n_i} \quad (9)$$

This is the kernel of the likelihood function to maximize. However, it is still cumbersome to differentiate and can be simplified a great deal further by taking its log. Since the logarithm is a monotonic function, and a maximum of the likelihood function will also be a maximum of the log-likelihood function and vice versa. Thus, taking the natural log (9) yields the log-likelihood function:

$$l(\beta) = \sum_{i=1}^N y_i \left(\sum_{k=0}^K x_{ik} \beta_k \right) - n_i \cdot \log(1 + e^{\sum_{k=0}^K x_{ik} \beta_k}) \quad (10)$$

To find the critical points of the log-likelihood function, set the first derivative with respect to each equal to zero. In differentiating (10), we note:

$$\frac{\partial}{\partial \beta_k} \sum_{k=0}^K x_{ik} \beta_k = x_{ik} \quad (11)$$

Thus (10) becomes:

$$\begin{aligned} \frac{\partial l(\beta)}{\partial \beta_k} &= \sum_{i=1}^N y_i x_{ik} - n_i \cdot \frac{1}{1 + e^{\sum_{k=0}^K x_{ik} \beta_k}} \cdot \frac{\partial}{\partial \beta_k} (1 + e^{\sum_{k=0}^K x_{ik} \beta_k}) = \sum_{i=1}^N y_i x_{ik} - n_i \cdot \frac{1}{1 + e^{\sum_{k=0}^K x_{ik} \beta_k}} \cdot e^{\sum_{k=0}^K x_{ik} \beta_k} \times \\ &\frac{\partial}{\partial \beta_k} \sum_{k=0}^K x_{ik} \beta_k = \sum_{i=1}^N y_i x_{ik} - n_i \cdot \frac{1}{1 + e^{\sum_{k=0}^K x_{ik} \beta_k}} \cdot e^{\sum_{k=0}^K x_{ik} \beta_k} \cdot x_{ik} = \sum_{i=1}^N y_i x_{ik} - n_i \pi_i x_{ik} \quad (12) \end{aligned}$$

The maximum likelihood estimate β can be found by setting, each of the K+1 in (12) to zero and solving for each β_k : By differentiating for the second time; thus:

$$\frac{\partial^2 l(\beta)}{\partial \beta_k \partial \beta_{k'}} = \frac{\partial}{\partial \beta_{k'}} \sum_{i=1}^N y_i x_{ik} - n_i \pi_i = \frac{\partial}{\partial \beta_{k'}} \sum_{k=i}^K -n_i x_{ik} \pi_i = - \sum_{k=i}^K n_i x_{ik} \frac{\partial}{\partial \beta_{k'}} \left(\frac{e^{\sum_{k=0}^K x_{ik} \beta_k}}{1 + e^{\sum_{k=0}^K x_{ik} \beta_k}} \right) \quad (13)$$

Solving (13) further by rules of differentiation:

$$\frac{d}{dx} \frac{e^{u(x)}}{1 + e^{u(x)}} = \frac{(1 + e^{u(x)}) \cdot e^{u(x)} \frac{d}{dx} u(x) - e^{u(x)} \cdot e^{u(x)} \frac{d}{dx} u(x)}{(1 + e^{u(x)})^2} = \frac{e^{u(x)} \frac{d}{dx} u(x)}{(1 + e^{u(x)})^2} = \frac{e^{u(x)}}{1 + e^{u(x)}} \cdot \frac{1}{1 + e^{u(x)}} \cdot \frac{d}{dx} u(x) \quad (14)$$

And (14) can now be written as:

$$\frac{\partial^2 l(\beta)}{\partial \beta_k \partial \beta_{k'}} = - \sum_{i=1}^N n_i x_{ik} \pi_i (1 - \pi_i) x_{ik'} \tag{15}$$

Having verified that, the matrix of second partial derivatives is negative definite and the solution is a global maximum, rather than a local maximum. Then we conclude that this vector contains the parameter estimates for which the observed data would have the highest probability of occurrence. This solution has to be numerically estimated using an iterative process, perhaps using Newton's method for solving nonlinear equations. Setting (12) equal to zero results in the system of a K+1 unknown variable β_k . Recall that, the Taylor polynomial of degree n for the point $x = x_0$ is defined as the first n terms of Taylor series for f

$$\sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \tag{16}$$

Provided that the first n derivatives of f at x_0 all exist $f(x_0)$ with $f(x) = 0$

$$f(x_0) + f'(x_0) \cdot (x - x_0) = 0 \tag{17}$$

Solving for (x) we have

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The value of x is the next approximation for the root. We let $x_1 = x$ and continue in the same manner to generate $x_2, x_3 \dots$, until the approximations converge. We write (12) as $l'(\beta)$. Let β^0 represent the vector of initial approximation for each β_k , and then the first step of Newton-Raphson can be expressed as:

$$\beta^{(1)} = \beta^{(0)} + [-l''(\beta^{(0)})]^{-1} \cdot l'(\beta^{(0)}) \quad (18)$$

Let μ be a column vector of length N with the element $\mu_i = n_i \pi_i$: $\mu_i = E(y_i)$ the expected value of y_i using matrix multiplication:

$$l'(\beta) = X^T(y - \mu) \quad (19)$$

Equation (19) is a column vector of length K + 1, whose elements as $\frac{\partial l(\beta)}{\partial \beta_k}$ derived in (11). Now, let W be a square matrix of order N with the element $n_i \pi_i (1 - \pi_i)$ on the diagonal and zeros everywhere else. Again, using matrix multiplication we verify that:

$$l''(\beta) = -X^T W X \quad (20)$$

(20) K + 1 by K + 1 square matrix. Now (18) can be written as:

$$\beta^{(1)} = \beta^{(0)} + [X^T W X]^{-1} \cdot X^T(y - \mu) \quad (21)$$

Continue applying (21) until there is essentially no change between the elements of β within iterations. At the point; The maximum likelihood estimates are said to be converged, (20) will hold the variance-covariance matrix of the estimate.

Classification accuracy

According to [Muhammad et al. \(2018\)](#), “The purpose of the classification is to assign a class to discover formerly unseen records as accurately as possible. If there is a group of records (called a training set) and each record contains a set of attributes, then one of the attributes is class [Chao and Wong \(2009\)](#), [Podgorelec and Maribor \(2005\)](#). The motive is to find a classification model for class attributes, where a test set is used to find out the accuracy of the model. The acknowledged figured set are divided into training and testing sets. The training set used to fabricate the model and testing set is used to authenticate it [Wang and Zhou \(2005\)](#), [Karegowda and Jayaram \(2009\)](#). classification practice consists of a training set that is analysed by a classification algorithm and the classifier or learner [Tang and Tseng \(2009\)](#). Model is represented in the composition of classification rules [Xue and Yanan \(2006\)](#)”. Testing data are used in the classification rules to estimate the accuracy. The learner model is represented in the form of classification rules or decision trees. The reference model was built by entry of 5 variables followed by removal of those with no significant partial correlation (R Statistic). The SPSS version 25 for Windows (10.1) was used for these analyses.

The Multi-layer perceptron

We used a common feedforward backpropagation multilayer perceptron (MLP) simulator developed in SPSS software package. The prediction method is based on the nonlinear weighted combination of input units (i.e. predictive variables) to predict one or more output units (i.e. outcome variable). The learning process is iterative and essentially consists in adjusting the weights to decrease the output error. The network was specified with one input layer (representing the five predictive variables), one hidden layer (including five hidden units) and one output layer (with one output unit representing a binary diabetic event). Several sensitivity analyses were performed to test how the prediction results could be influenced by the variations of learning parameters and to elicit the most optimized network. These parameters refer to the architecture of the network (number of hidden units), the method of internal validation (number of iterations and data-splitting processes), the options of data pre-treatment (i.e. normalization of inputs), the activation function for hidden units, and the “Score Threshold” used by the system to classify a case from its predicted probability.

The architecture used for the MLP is:

Input Layer: $J_0 = P$ units or variable; $a_{0:1}, \dots, a_{0:5}$;

with $a_{0:j} = x_j$,

and $J_i =$ Number of units in layer i , $a_{i:j}$ unit i of layer j

i th hidden layer : J_i units, $i = 1 \dots 5$, $a_{i:1}, \dots, a_{i:j}$ with $a_{i:k} = \gamma_i(C_{i:k})$

and $C_{i:k} =$

$\sum_{j=0}^{J_i-1} w_{i:j,k} a_{i-1:j}$ where $a_{i-1:0} = 1$. $\gamma_i(C) =$ activation function for layer i

Output layer: : $J_I = R$ units, $a_{i:1}, \dots, a_{I:j}$ with $a_{i:k} = \gamma_i(C_{i:k})$

and $C_{i:k} =$

$\sum_{j=0}^{J_i-1} w_{I:j,k} a_{I-1:j}$ where $a_{i-1:0} = 1$.

The activation function of the hidden layer is the hyperbolic tangent given as

$$\gamma(C) = \tanh(c) = \frac{e^c - e^{-c}}{e^c + e^{-c}}$$

The activation function of the output layer is the sigmoid function given as

$$\gamma(C) = \frac{\exp(C_k)}{\sum \exp(C'_j)}$$

The algorithm involved in MLP are as follows

1. Start with an initial network of k hidden units. The default is $k = \min (g (R, P), 20, h (R, P))$, where,

$$g(R, P) = \begin{cases} \frac{4.5}{P+R} & R < 5, P \geq 8 \\ 0.5 + 0.5(P + R) & \text{otherwise} \end{cases}$$

and $h(R,P)=[M-R/P+R+1]$. If $k < k_{min}$, set $K = k_{min}$. Else if $K > k_{max}$, Set $k = k_{max}$.

2. If $K > k_{min}$, Set DOWN = TRUE. Else if training error ratio > 0.01, DOWN = FALSE. Else stop and report the initial network.

3. If DOWN=TRUE, remove the weakest hidden unit (see below); $k=k-1$. Else add a hidden unit; $k=k+1$.

4. Using the previously fit weights as initial weights for the old weights and random weights for the new weights, train the old and new weights for the network once through the alternated simulated annealing and training procedure (steps 3 to 5) until the stopping conditions are met.

5. If the error on test data has dropped:

If DOWN=FALSE, if $k < k_{max}$ and the training error has dropped but the error ratio is still above 0.01, return to step 3. Else if $k > k_{min}$, return to step 3. Else, stop and report the network with the minimum test error.

Else if DOWN=TRUE, if $|k-k_0| > 1$, stop and report the network with the minimum test error. Else if training error ratio for $k=k_0$ is bigger than 0.01, set DOWN=FALSE, $k=k_0$ return to step 3. Else stop and report the initial network.

Else stop and report the network with the minimum test error.

If more than one network attains the minimum test error, choose the one with fewest hidden units.

If the resulting network from this procedure has training error ratio (training error divided by error from the model using average of an output variable to predict that variable) bigger than 0.1, repeat the architecture selection with different initial weights until either the error ratio is ≤ 0.1 or the procedure is repeated 5 times, then pick the one with smallest test error.

Using this network with its weights as initial values, retrain the network on the entire training set.

Confusion Matrix

A confusion matrix table is a table with 2 X 2 rows and columns that report the number of false positive, false negative and true positive and negative. It displays further analysis relation to classification and aspect of machine learning. [Stehman \(1997\)](#) refers to confusion matrix or error matrix as a specific table layout that visualizes algorithm and performance of supervised learning.

3. RESULTS AND DISCUSSIONS

3.1 LOGISTIC REGRESSION

Analysis showed that the average age of all the cases involved in the study is 35.78 with a standard deviation of 8.73. The average insulin level is 92.0054 mu. While the average glucose level of the study was found to be 6.5230, with a weight 84.145 kg.

It is seen that there is a positive correlation between the class variable and all the predictor variables: glucose, insulin, age and weight, except pressure variable.

In other words, the higher the values on each of the variables, the more likely the patient is classified 1, that is diabetic. The negatively correlated pressure variable means the opposite. It can also be observed that the two variables glucose and weight have the best relationship with the dependent variable that is class.

It is also observed that, without predictor variables about 247 out of 553 cases will be classified as non-diabetic with overall percentage of 55.3 of the model correctly classifying cases.

Also, glucose and weight have a significant value of 0.0 which is less than 0.05 and that means both variables have a good predictive ability for a case. Pressure, insulin and age have significance of 0.714, 0.103 and 0.663 respectively. All of the three variables have significance greater than 0.05 and that means they are not significant in predicting the outcome of case.

The Omnibus Test showed a chi-square value of 553.693 for the model with a p-value of 0.000. A significance level that is less than 0.05 indicate the model is good for predicting the outcome of a case.

The Nagelkarke R squared value showed an 84.7% variance in the dependent variable explained for by the independent variables.

Hosmer and Lemeshow Test on had a significance value of 0.544 that is greater than 0.05 and that indicates again that the model has a good predictive capacity.

This is further confirmed when the outcome of the model is fitted to actual outcomes. Observing class 1, at step 10, out of 58 already classified cases as diabetic, the model predicted almost all 58 cases correctly.

It is also seen that glucose has a coefficient value of 3.343 with an odd ratio of 28.292 which means for a case with a high glucose value, it is 28.292 times likely to be classified as diabetic.

Also, it is seen that glucose and weight have the highest odd ratio meaning a case with a high value in either glucose or weight will have a chance of being classified diabetic.

Table 1 Effectsof each variable on the logistic regression model.

| | S.E. | Wald | Df | Sig. | Exp(B) | |
|-----------------|---------|-------|--------|------|--------|--------|
| glucose | 3.343 | .345 | 93.772 | 1 | .000 | 28.292 |
| pressure | -.013 | .015 | .720 | 1 | .396 | .987 |
| insulin | .008 | .004 | 3.571 | 1 | .059 | 1.008 |
| age | .019 | .021 | .809 | 1 | .368 | 1.019 |
| weight | .212 | .024 | 79.390 | 1 | .000 | 1.236 |
| Constant | -39.113 | 4.424 | 78.171 | 1 | .000 | .000 |

From **Table 1** , the constant term of the logistic regression equation is found to be -39.113. The coefficient of glucose, pressure, insulin, age and weight are 3.343, -0.13, 0.008, 0.019, 0.212 respectively. Thus, the logistic regression equation is therefore given as

$$\pi_i = \frac{e^{-39.113+3.343x_1-0.13x_2+0.008x_3+0.019x_4+0.212x_5}}{1 + e^{-39.113+3.343x_1-0.13x_2+0.008x_3+0.019x_4+0.212x_5}}$$

$i = 0, 1$

From **Table 2** , a case has an 89.9 % chance of being correctly classified as being not diabetic. Also, it is observed that a case has a 93.1% of being correctly classified as being diabetic. The overall model has a 91.7% classification accuracy.

As seen from **Table 2** , there is presence of collinearity among some variables. This led to the following variables excluded from the model: Pressure, Age and Insulin. Also, from Table 4 the variables have significance level of 0.714, 0.103 and 0.663 for Pressure, Insulin and Age respectively. This means that the variables have insignificant effect on the model. The analysis is run again without the above variables. It is observed that the reduced model correctly classifies a non-diabetic patient with 89.1% and classifies a diabetic patient with 91.8% accuracy. The model generally classifies with a total of 90.6%.

The reduced model is therefore given as

$$\pi_i = \frac{e^{-38.725+3.315x_1+0.211x_5}}{1 + e^{-38.725+3.315x_1+0.211x_5}}$$

$i = 0, 1$

Table 2 Confusion Matrix.

| Observed | Predicted | |
|----------|-----------|--------------------|
| | Class | Percentage Correct |

Continued on next page

Table 2 continued

| | | 0 | 1 | |
|-------|--------------------|-----|-----|------|
| class | 0 | 222 | 25 | 89.1 |
| | 1 | 21 | 285 | 91.8 |
| | Overall Percentage | | | 90.6 |

4. MULTI LAYER PERCEPTRON

Multi-layer Perceptron (MLP) was applied. In MLP, the Input layer has 5 factors with 315 units excluding the bias unit as seen from Table 2. The hidden layer has 2 units excluding the bias unit. The Hyperbolic tangent was the activation function in the hidden layer. The output layer has 1 dependent variable which is 'class', with 2 units. Softmax was the activation function. Cross-entropy was the error function used.

368 cases were used in the training sample. the network weights that corresponded to the lowest mean squared error on the validation set were used for evaluation on the test data.

The test data has 115 cases and 61 hold-out cases. 9 cases had factors that do not occur in the training sample, as a result they were excluded from the analysis.

For class 0 in the training sample, the network has 97.7 % correct classification and 99.5% correct classification for class 1. The testing sample has 84.1% correct classification and 84.5 % correct classification for classes 0 and 1 respectively.

| Table 3 Multi-Layer Perceptron Network Information | | | |
|--|--|---|--------------------|
| Network Information | | | |
| Input Layer | Factors | 1 | glucose |
| | | 2 | pressure |
| | | 3 | insulin |
| | | 4 | age |
| | | 5 | weight |
| | Number of Units ^a | | 315 |
| Hidden Layer(s) | Number of Hidden Layers | | 4 |
| | Number of Units in Hidden Layer 1 ^a | | 1 |
| | Activation Function | | Hyperbolic tangent |
| Output Layer | Dependent Variables | 1 | class |
| | Number of Units | | 2 |
| | Activation Function | | Softmax |
| | Error Function | | Cross-entropy |

It can be seen from Table 4, when Logistic regression model is compared with Multi-Layer Perceptron that the percentage of correctly classifying a diabetic patient as diabetic in LR is 91.8% and 88.6% in MLP. Again, in LR, the percent correctly classifying a non-diabetic patient as non-diabetic is 89.1% while in MLP it is 93.2%. The

Table 4 MLP Classification

| | Actual Group | No. of cases | Predicted Group Membership | | Percent Correct |
|-----------------------|-----------------|--------------|----------------------------|-----------------|-----------------|
| | | | Diabetic(1) | Non-diabetic(0) | |
| Train- ing | Diabetic(1) | 192 | 191 | 1 | 88.60% |
| | Non-diabetic(0) | 176 | 4 | 172 | 93.20% |
| | Overall | 71 | 60 | 11 | 91.0% |
| Testing | Diabetic(1) | 42 | 7 | 37 | 93.0% |
| | Non-diabetic(0) | | | | 93.0% |
| | Overall | | | | 93.0% |

LR generally classifies with 90.6% accuracy while the MLP generally classifies with 91% accuracy.

Table 5 Comparison of MLP with LR in terms of classification

| | MLP | LR |
|------------------------|--------|--------|
| Diabetic(1) | 88.60% | 91.80% |
| Non-diabetic(0) | 93.20% | 89.10% |
| overall | 91.00% | 90.60% |

5. CONCLUSIONS AND RECOMMENDATIONS

The study was carried out to compare the classification power of Logistic Regression and Multi-layer perceptron. 553 records of data collected was on diabetic patients who were tested at Federal Medical Center, Yola. The variables measured include: the glucose level of each patient, diastolic pressure, insulin level, weight of each patient and their ages. The task was to see which of the two techniques classifies better.

First, at the implementation stage, we chose to evaluate the methods at their best performance, i.e. after optimization of the modeling specifications. This required to understand the meaning of each learning parameter and to test its influence on final results.

SPSS was used to run the analysis for both techniques. The neural networks tab was used on the software. In Multi-Layer Perceptron, 70% of the data was used to train the network, 20% was used for testing the trained network and 10% was used for the hold out sample. The Binary logistic regression tab on SPSS was used to fit a logistic regression model on the data.

The logistic regression model has a correct classification percentage of 90.6%. The Multilayer Perceptron, on the other hand has a correct classification as diabetic

with 91.0% of correctly classifying a case.

When comparing the performance of Logistic regression and MLP on the diabetes data as a case study, both had good classification power. The overall classification rate for both was good, and either can be helpful in classifying the class membership of women that are diabetic. The MLP exceed the Logistic Regression Model in the overall correct classification rate.

6. ACKNOWLEDGEMENTS

We express our appreciation to Dr. S. S Abdulkadir, Department of Statistics and Operations research, Modibbo Adama University of Technology, Yola, Adamawa State, Nigeria, for guidance throughout the study. We also appreciate Federal Medical Center, Yola, for providing us with data to carry out the study.

REFERENCES

- Alan, A. (2002). *Categorical Data Analysis*. (2nd Edition ed.). University Of Florida. New York: Wiley.
- Chao, S., & Wong, F. (2009). An Incremental Decision Tree Learning Methodology Regarding Attributes In Medical Data Mining. In *Proceedings Of The Eighth International Conference On Machine Learning And Cybernetics*.
- Hosmer, D., & Lemeshow, S. (2000). *Applied Logistic Regression*. In and others (Ed.), (2nd Ed ed., pp. 1–2). Wiley And Sons Inc.
- Hosmer, D. W., Lemeshow, S., & Klar, J. (1988). A Goodness-Of- Fit Test For The Multiple Regression Model. *Communications In Statistics*, 10, 1043–1069.
- Joaquim, P., & Sá, M. D. (2007). *Applied Statistics Using SPSS, STATISTICA, MATLAB And R*. (2nd Edition ed.). Berlin: Springer-Verlag.
- Karegowda, A. G., & Jayaram, M. A. (2009). Cascading GA & CFS For Feature Subset Selection In Medical Data Mining., *IEEE*, 1–4.
- Lewicki, P., & T, H. (2006). *Statistics: Methods And Applications: Comprehensive Reference For Science, Industry, And Data Mining*. In and others (Ed.), . Statsoft, Inc.
- Muhammad, M. U., Jiadong, R., Sohail, M. N., Irshad, M., Bilal, M., & Osi, A. A. (2018). A Logistic Regression Modeling On The Prevalence Of Diabetes Mellitus In The North Western Part Of Nigeria. *Benin Journal Of Statistics*, 1, 1–10.
- Podgorelec, V., & Maribor, H. M. (2005). Improving Mining Of Medical Data By Outliers Predictions., *IEEE*, 1–6.
- Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62, 77–89. Retrieved from [https://dx.doi.org/10.1016/s0034-4257\(97\)00083-7](https://dx.doi.org/10.1016/s0034-4257(97)00083-7)
- Tang, P. H., & Tseng, M. H. (2009). Medical Data Mining Using BGA & RGA For Weighting Of Features In Fuzzy KNN Classi_Cation., 5, 1–6.
- Wang, S., & Zhou, G. G. (2005). Application Of Fuzzy Clusters Analysis For Medical Imagedata Mining., 2, 1–6.
- Wehrens, R. (2010). *Chemometrics With R Multivariate Data Analysis In The Natural Sciences And Life Sciences*. In and others (Ed.), . Springer.
- Xue, W., & Yanan, S. Y. (2006). Research And Application Of Data Mining In Traditional Chinese

Medical Clinic Diagnosis., 4, 1-4.