Predicting Diabetes Using Perceptron

Shivangi
The University of Adelaide
Adelaide, South Australia, Australia
shivangi@student.adelaide.edu.au

Abstract

Classification of diabetes is an important task in medical diagnosis, whereby early diagnosis enhances the recovery chances of a patient. The paper presents the development of the Single-layer Perceptron model in diagnosing whether a patient has diabetes or not. The available dataset consists of Skin Thickness, Insulin, BMI,Pregnancies, Blood Pressure, Diabetes Pedigree Function, Age, Glucose, and Outcome, which includes data from 768 women aged between 21 and 81. In this paper, we show that this simple Perceptron achieves a good accuracy of 76.62%.

1. Introduction

Diabetes classification has been addressed in many works using machine learning methodologies. For instance, in Darolia and Chhillar, Random Forest, Logistic Regression, and ANN are compared for the prediction of diabetes classification, achieving success rates of 76.1%, 74.75%, and 77.1%, respectively. Another approach implemented K-Nearest Neighbors with modified distance metrics to achieve an accuracy of 83% in diabetes classification.

These articles highlight the success of complex models, but they also indicate the relevance of simpler and more interpretable models, such as the Perceptron, especially for smaller datasets or linearly separable problems. Diabetes is one of the most widespread diseases worldwide, and if untreated, it can lead to serious complications such as heart conditions and kidney failure. Early detection is crucial, and machine learning models can significantly aid in analyzing medical datasets for predicting outcomes.

In this work, we focus on a simpler yet powerful model: the single-layer Perceptron. The single-layer Perceptron is often seen as unfavorable compared to more complex models such as Multilayer Perceptrons (MLP) but can serve as a strong baseline for linearly separable problems. We implemented a from-scratch single-layer Perceptron and evaluated its performance using accuracy on the Pima Indians Diabetes dataset, containing health metrics predictive of diabetes.

2. Method

This section provides an overview of the materials and methods used in the study. A detailed breakdown of the dataset, methodology, and approach will be outlined in subsequent subsections.

2.1. Dataset

The dataset consists of nine attributes, out of which eight are predictors, and a binary target variable labeled 'Outcome,' where '1' indicates the presence of diabetes and '0' indicates no diabetes. It includes data from 768 women, all of whom were over 21 years of age. We also need to be aware about the important feature.. Important feature helps us indicate which variables actually determine the forecasts of the model, and based on these facts, we can discard the less important ones, doing so would result in the simplification of the model and presumably lead to better performance thereof. Feature importance can be quantified by considering absolute values of weights assigned to each one of the features of the input by the perceptron model. This analysis helps us prioritize certain features responsible for the model's predictions and provides insight into both intrinsic and extrinsic relationships in the data. Figure 1 shows the weight values of each feature involved, hence their relative importance in the classification task of diabetes.

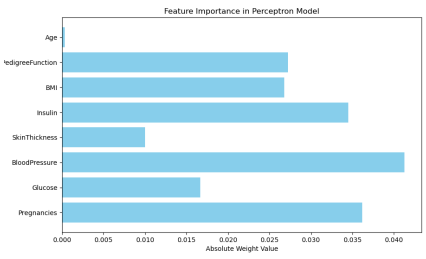

Figure 1. Feature Importance in the Perceptron Model

Feature	Range	Type
SkinThickness	0,99	Numerical
Insulin	0,846	Numerical
BMI	0,67.1	Numerical
Pregnancies	(0,17)	Numerical
Glucose	0,199	Numerical
BloodPressure	0,122	Numerical
DiabetesPedigreeFunction	0.078,2.42	Numerical
Age	21,81	Numerical
Outcome	0 or 1	Categorical

Table 1. Features of the Pima Indians Diabetes Dataset

2.2. Perceptron Algorithm

The simplest Artificial Neural Network structures involve Single-Layer-Perceptron. A Perceptron can be defined as a linear threshold device that calculates the weighted sum of the coordinates of the pattern vector, compares the value with a threshold, and outputs +1 or -1 if the threshold is reached. The threshold is labeled as an activation function:

$$z = \mathbf{w} \cdot \mathbf{x} + b$$

Where:

- $\mathbf{w}$ represent the weight vector.
- $\mathbf{x}$ represent the feature vector(or input data).
- b represents bias.

Step function is used as the activation function:

$$f(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

During training, the weights are updated based on the error in predictions:

$$\mathbf{w} = \mathbf{w} + \eta(y - \hat{y})\mathbf{x}$$

Where η is the learning rate, y is the true label, and $\hat{y}$ is the predicted label.

2.3. Implementation

Here Perceptron is implemented in Python, training it for 1000 iterations with a learning rate of 0.01. The dataset was split into two sets named as training(contains 80% data) and testing set(contains 20% data). The features were standardized using the **StandardScaler** from Scikit-learn to ensure that all features were on a similar scale.

2.4. Evaluation and Analysis

We evaluated the performance of our model on the test set for different metrics, including accuracy, precision, recall, and execution time. We used a confusion matrix to get an idea about the effectiveness of the classification model through the visualization of counts for true positives, true negatives, and false positives and false negatives. From figure 2, it can be seen that the model was more accurate with regards to non-diabetic data rather than diabetic data.

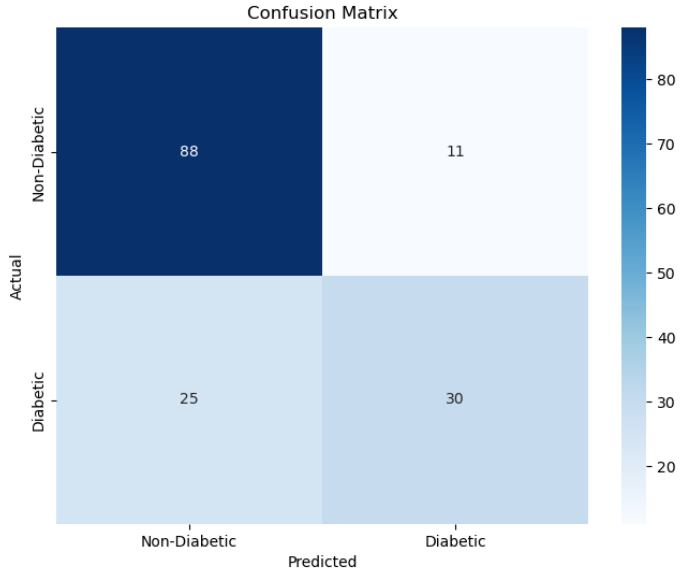

Figure 2. Confusion Matrix for the Perceptron Model

In order to get a better estimate of our model, we have used the ROC curve-which plots the balance between TPR vs. FPR at different threshold values. The performance measure that tells us something about the power of the model in distinguishing between the two classes is given by the area between the ROC and the uniform distribution curve. Fig. 3 And the ROC curve of the Perceptron model: From this ROC curve, we know that the optimum AUC shows better discrimination between diabetic and non-diabetic instances. From the ROC curve, we can choose an optimal cut-off where we can balance sensitivity and specificity to effectively evaluate the model or compare it with other models; hence, it gives critical insights in making an informative decision.

3. Code

The code and the data is available athttps://github.com/shivangi-crypto/Diabetes-Prediction-using-Perceptron.git

4. Conclusion

The single-layer Perceptron, while being a very simple, foundational algorithm, could still deliver quite reasonable results in the case of linear classification tasks, such as diabetes prediction. The weakness of the Perceptron when it

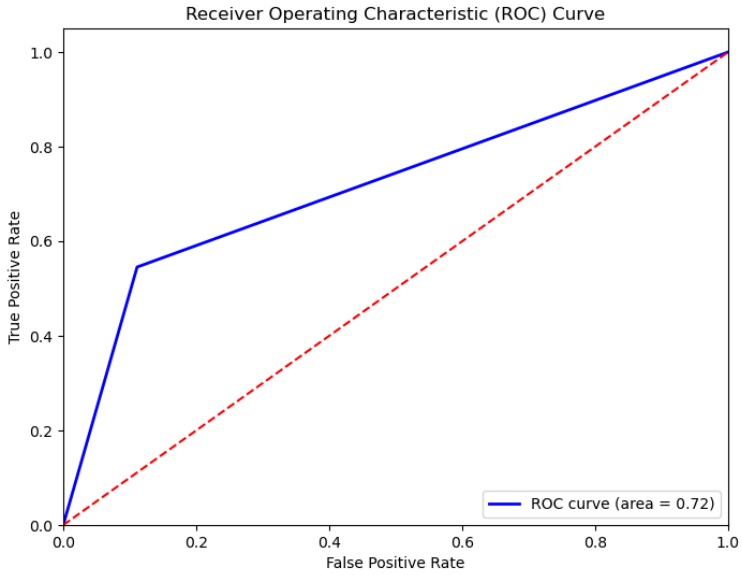

Figure 3. ROC Curve for the Perceptron Model

comes to modeling non-linear relationships cascades to performance metrics. Further work may look ahead at the implementation of more complex models or the investigation of other feature spaces in a view to improving the predictive accuracy.