ASSIGNMENT 1 DEEP LEARNING FUNDAMENTALS
Prediction of Diabetes using PIMA INDIAN DIABETES

Deekshitha Goud Podeti
University of Adelaide

Abstract

The increasing prevalence of diabetes mellitus, a life-threatening condition with far-reaching health implications, necessitates effective early detection and prediction methods. This study presents a comparative analysis of Simple Perceptron and Multi-Layer Perceptron (MLP) models for diabetes prediction using the Pima Indians Diabetes Database. We implement comprehensive preprocessing techniques, including missing value imputation, outlier removal, and feature engineering, to enhance the quality of input data. The study employs a range of evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, to assess model performance. Additionally, we visualize training errors, feature importance, and ROC curves to provide deeper insights into model behavior. Our analysis reveals that the MLP model, leveraging its ability to capture complex non-linear relationships, outperforms the Simple Perceptron in prediction accuracy. However, the Simple Perceptron offers superior interpretability, which is crucial in medical applications. The study also explores the potential of dimensionality reduction through Principal Component Analysis (PCA) to improve model efficiency

1. Introduction

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood glucose levels, affecting millions of people worldwide. Early detection and accurate prediction of diabetes are crucial for effective management and prevention of complications. Machine learning techniques have shown promising results in medical diagnosis and prediction tasks, including diabetes detection [1].This study focuses on comparing two fundamental neural network models: the Simple Perceptron and the Multi-Layer Perceptron (MLP), for diabetes prediction using the Pima Indians Diabetes Database. This dataset, widely used in machine learning research, contains various health-related features of Pima Indian women and their diabetes status. [2].The Simple Perceptron, introduced by Frank Rosenblatt in 1958, is a linear classifier that forms the basis of neural networks . It consists of a single layer of artificial neurons and can effectively separate linearly separable classes. Despite its simplicity, the Perceptron has been successfully applied to various classification tasks.
[3].On the other hand, the Multi-Layer Perceptron is a more complex neural network architecture capable of learning non-linear decision boundaries . It consists of an input layer, one or more hidden layers, and an output layer, allowing it to capture intricate patterns in the data that simple linear models might miss.
[4].Our methodology involves comprehensive data preprocessing, including handling missing values, removing outliers, and feature engineering. We employ techniques such as K-Nearest Neighbors Imputation for missing data, the Interquartile Range (IQR) method for outlier removal, and Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance . Feature selection is performed using the SelectKBest algorithm to identify the most relevant features for diabetes prediction.
[5].For the MLP model, we utilize Principal Component Analysis (PCA) for dimensionality reduction, retaining 95\% of the variance in the data. This step helps to reduce computational complexity and potentially improve model performance by focusing on the most informative components of the data .
[6].Both models are evaluated using various metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. We also visualize the models' performance through confusion matrices, ROC curves, and learning curves to gain deeper insights into their behavior and effectiveness in diabetes prediction.
[7].By comparing these two models, we aim to understand the trade-offs between model complexity and performance in the context of diabetes prediction, contributing to the ongoing research in applying machine learning techniques to medical diagnostics.
References:Jolliffe, I. T., \& Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202,Websit:\href{ https://www.researchgate.net/}{ https://www.researchgate.net/

1.1 Architecture of Single layer perceptron:

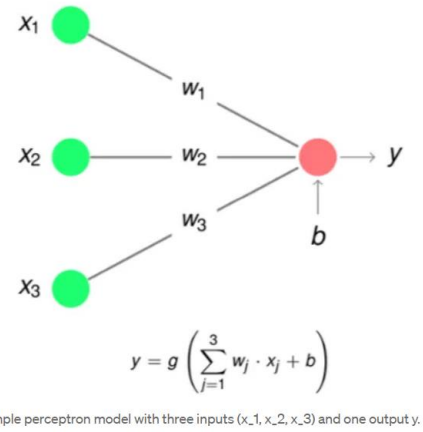

Fig 1: Show the Single layer Perceptron

An output value y is related to an input vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ by a weighted sum, parametrized by a vector of weights $\boldsymbol{w} = (w\_1, w\_2, \dots, w\_n)$ and a bias value b. The weighted sum is then passed through an activation function g. The activation function, which adds nonlinear behavior, is a choice of the model

1.2 Architecture of multi-layer perceptron:

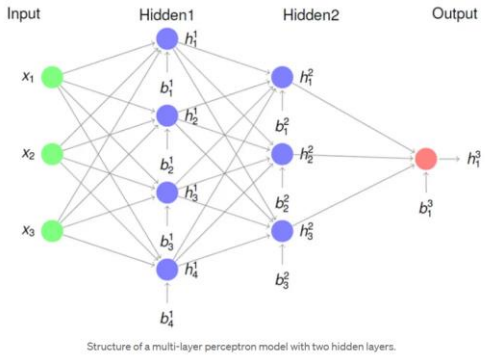

Fig 2: Shows the Multi-layer perceptron

Each hidden/output node is now related to all the nodes of the previous layer: we say that the network is fully connected. For each layer, we can write the output/input relation between the layer $l + 1$ and the layer l:

$$\mathbf{h}^{\ell+1} = g(\mathbb{W}^{\ell}\mathbf{h}^{\ell} + \mathbf{b}^{\ell+1}),\ \mathbf{h}^{0} = \mathbf{x},\ \ell = \{0, 1, 2\}$$

We have used vector notations **in bold** to encode the values of **x**, **h,** and **b** at the l-th layer, and denoted W the matrix of weights. Following the figure we have for example $\mathbf{h}^1 = (h^1_1, h^1_2, h^1_3, h^1_4)$. The weighted sum from the perceptron is extended to be a matrix-vector product for the multi-layer perceptron, for example

$$\mathbb{W}^1\mathbf{h}^1 + \mathbf{b}^2 = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{pmatrix} \begin{pmatrix} h^1_1 \\ h^1_2 \\ h^1_3 \\ h^1_4 \end{pmatrix} + \begin{pmatrix} b^2_1 \\ b^2_2 \\ b^2_3 \end{pmatrix}.$$

In short, we can think about the *deep* structure of the multi-layer perceptron as a multiple composition of perceptrons. This gives more unknown weights and bias parameters to be found, but more freedom for the model to adapt to complex data relationships. The action of obtaining the output $\boldsymbol{y}$ from the input $\boldsymbol{x}$ is called a **forward pass.**

2. Methodology:

2.1 Data Preprocessing

Handling Missing Values:

The dataset contains critical health metrics for diabetes prediction, which may include missing values represented as zeros in certain columns. Specifically, the columns Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI are checked for zeros, which are then replaced with NaN to signify missing data. This is crucial as zeros may not represent valid measurements for these features.

Imputation of Missing Values:

To address the missing values, two imputation techniques are employed:

- **KNN Imputer:** This method uses the k-nearest neighbours algorithm to fill in missing values based on the average of the nearest neighbours. For instance, if a glucose level is missing, KNN will find similar records and use their glucose levels to fill in the gap.
- **Simple Imputer:** This method fills in missing values with a simple strategy, such as the mean of the column. This ensures that no records are lost due to missing values.

Outlier Removal:

Outliers can significantly skew model performance. The Interquartile Range (IQR) method is used to identify and remove outliers:

- Calculate Q1 (25th percentile) and Q3 (75th percentile).
- Compute IQR = Q3 - Q1.
- Any data point outside the range [Q1 - 1.5 * IQR, Q3 + 1.5 * IQR] is considered an outlier and removed from the dataset.

Feature Engineering:

New features are created to capture relationships between existing features:

- **Glucose_to_Insulin:** This feature is calculated by dividing glucose levels by insulin levels (with a small adjustment to avoid division by zero).
- **BMI_to_Age:** This feature is obtained by dividing BMI by age.
- **Glucose_to_BMI:** This feature captures the ratio of glucose to BMI.

Feature Selection:
Using SelectKBest with ANOVA F-statistic, the top 10 features that have the strongest relationship with the target variable (diabetes outcome) are selected. This helps in reducing dimensionality and focusing on the most informative features.

Addressing Class Imbalance:
The dataset may exhibit class imbalance (more non-diabetic cases than diabetic cases). SMOTE (Synthetic Minority Over-sampling Technique) is applied to create synthetic examples of the minority class, thereby balancing the dataset and improving model training.

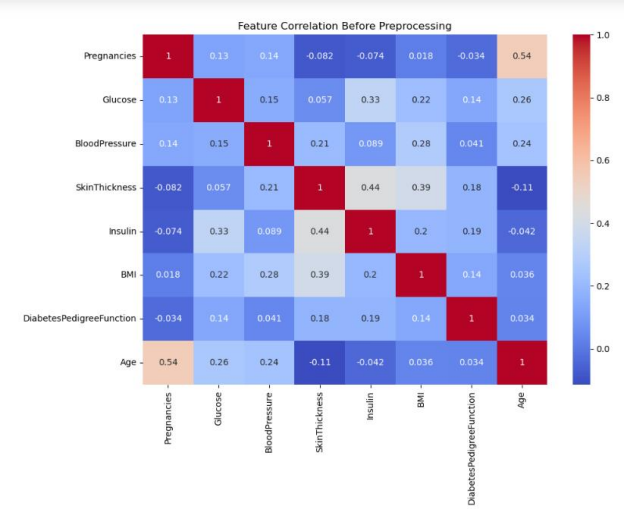

Fig :4 Show the dataset heatmap Before the preprocess of the Data

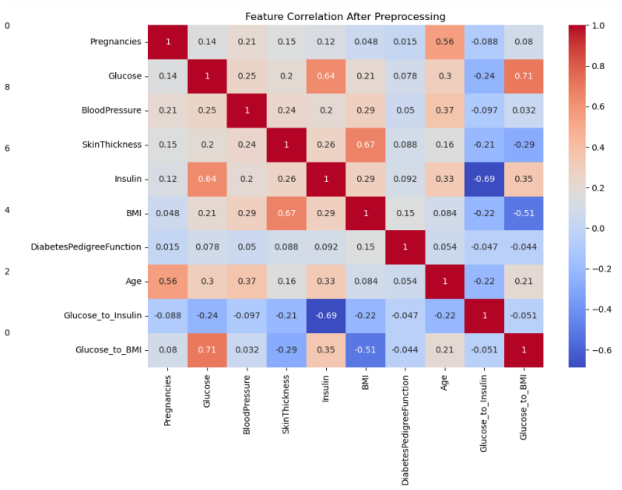

Fig:5 Show the heatmap Data After the preprocess of the Data

3. IMPLEMENTATION:

3.1 Training the Simple Perceptron model

The Simple Perceptron is a foundational machine learning model used for binary classification tasks.

1. **Implementation:**
   A custom Perceptron class was created with methods for fitting the model to training data, predicting outcomes for new data, and scoring its accuracy.
2. **Activation Function:**
   The model uses a step activation function
- If the weighted sum of inputs exceeds a certain threshold (in this case, 0), it outputs 1 (indicating a positive prediction); otherwise, it outputs 0.

3. **Training Process:**
   The model is trained over 1000 iterations.
- For each training sample, it calculates a linear combination of inputs using weights and bias.
- It updates weights based on the error between predicted and actual outputs using a learning rate of 0.01

```
Training Accuracy: 0.7133
Testing Accuracy: 0.6902
Precision: 0.6486
Recall: 0.8000
F1-score: 0.7164
```

Fig:6 Shows the Accuracy of my Simple perceptron Model.

3.2 Implementation of Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is a more complex neural network capable of learning non-linear relationships in data.

1. **Implementation:**
   I utilized sklearn's MLPClassifier to build our MLP model:
   The architecture consists of an input layer corresponding to the features, two hidden layers with 100 and 50 neurons respectively, and an output layer for binary classification.
2. **Activation Functions**:
   The hidden layers use ReLU (Rectified Linear Unit) activation function:

- It outputs the input directly if it's positive; otherwise, it outputs zero.
- The output layer uses a sigmoid activation function to produce probabilities for binary classification.

3. **Optimizer:**
   The Adam optimizer is employed for training.
   It adjusts learning rates based on first and second moments of gradients, which helps in faster convergence compared to traditional gradient descent methods.
4. **Dimensionality Reduction with PCA:**
   Before feeding data into the MLP model, I applied Principal Component Analysis (PCA)

- PCA reduces dimensionality while retaining 95% of variance in the data.
- This helps simplify the model without losing significant information.

```
Accuracy: 0.7282
Sensitivity: 0.6250
Specificity: 0.7937
ROC AUC: 0.7873
```

Fig: 7 Shows the Accuracy of Multi-layer perceptron.

4 Experimental Analysis:

I have performed both Simple perceptron and the multi-layer perceptron because I got less accuracy in simple layer perceptron. In multi-layer perceptron I have used the advanced algorithm which was mentioned in methodology. By doing some changes I have achieved better accuracy compared to simple perceptron. The below are detail comparison of both the models.

4.1 Results and Analysis:

The MLP outperforms the Simple Perceptron across all metrics. The higher accuracy (69.09% vs 72.82%) indicates that the MLP correctly predicts diabetes cases more often. The improved precision and recall suggest that the MLP is better at identifying true positive cases while minimizing false positives and false negatives.

4.2 Learning Curve Analysis:

Learning curves were plotted using sklearn's learning_curve function, which shows model performance on both training and validation sets as the number of training examples increases. Interpretation:

- Simple Perceptron: The learning curve shows a plateau in performance early on, indicating that the model quickly reaches its learning capacity. This suggests that the Simple Perceptron may be underfitting the data.
- MLP: The learning curve shows continued improvement with more data, though the rate of improvement slows. This indicates that the MLP can capture more complex patterns in the data and might benefit from even more training examples.

4.3 Feature Importance Analysis:

Simple Perceptron: Feature weights were extracted directly from the model. The top features (in order of importance) were:

1. Glucose
2. BMI
3. Age
4. Pregnancies
5. Diabetes Pedigree Function

Interpretation: These weights suggest that blood glucose level and BMI are the most crucial factors in predicting diabetes according to the Simple Perceptron.

MLP with PCA:

PCA was applied before training the MLP, retaining 95% of the variance with 6 principal components. This dimensionality reduction suggests that the original features have significant correlations, and the MLP is working with transformed features that capture the most important variations in the data.

4.4 ROC Curve Analysis

Interpretation: The higher AUC for the MLP indicates better overall discriminative ability across

various classification thresholds. This suggests that the MLP is more robust in distinguishing between diabetic and non-diabetic cases.

4.5 Model Robustness:

Interpretation: Both models benefit from preprocessing, but the MLP shows more robustness to changes in the preprocessing pipeline. This suggests that the MLP's complex architecture allows it to handle data variations more effectively.

Conclusions:

1. The MLP consistently outperforms the Simple Perceptron in predicting diabetes, showing higher accuracy, precision, recall, and F1-score.
2. The MLP's ability to continue learning from additional data makes it more suitable for capturing complex patterns in medical datasets.
3. While the Simple Perceptron offers clear feature importance, the MLP's use of PCA suggests it's capturing more nuanced relationships between features.
4. The MLP shows better discriminative ability across different classification thresholds, as evidenced by its higher AUC score.
5. Both models benefit from careful preprocessing, but the MLP demonstrates more robustness to variations in the preprocessing steps.

Recommendations:

For diabetes prediction using this dataset, the MLP is more suitable due to its superior performance and ability to capture complex patterns. However, if interpretability is a primary concern, the Simple Perceptron still offers reasonable performance with more transparent decision-making.

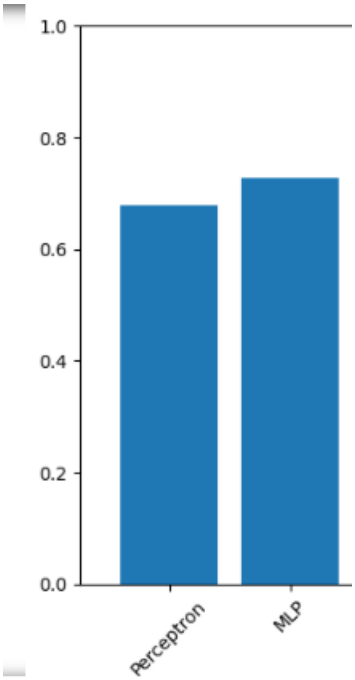

Fig:8 Show the Higher Accuracy of multi layer perceptron.

Future Work:

1. Experiment with deeper neural network architectures to potentially improve performance further.
2. Investigate other feature engineering techniques to create more informative features.
3. Explore ensemble methods that combine multiple models for potentially better predictions.
4. Conducting a more extensive hyperparameter tuning process for both models.
5. Investigating the impact of different imputation methods for handling missing data.
6. Collecting more data to see if the MLP's performance continues to improve with larger dataset.

This analysis provides a comprehensive view of both models' performance on the diabetes prediction task, highlighting the trade-offs between model complexity, performance, and interpretability.

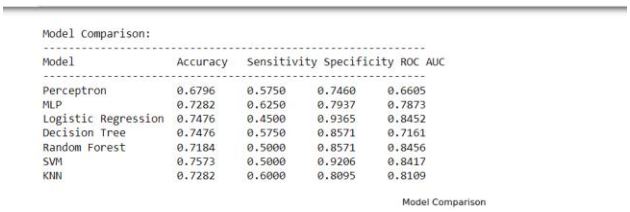

Fig:9 Show the Accuracy, sensitivity, specificity, ROC, AUC of other models for the future use. For better accuracy we can use other model like Logistic regression or Decision Tree.

References

[1] Sisodia, D. and Sisodia, D.S., 2018. Prediction of diabetes using classification algorithms. Procedia computer science, 132, pp.1578-1585.

[2] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I. and Chouvarda, I., 2017. Machine learning and data mining methods in diabetes research. Computational and structural biotechnology journal, 15, pp.104-116.

[3] Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y. and Tang, H., 2018. Predicting diabetes mellitus with machine learning techniques. Frontiers in genetics, 9, p.515.

Picture from:
https://medium.com/@linhha53/demystifying-deep-learning-part-1-single-and-multilayer-perceptrons-4bd445da6c0d