**Course: Introduction to Deep Learning**

**Diabetes Prediction**

**Exploratory Data Analysis**

**Student’s Name: Saja Musa, ID: 308424043**

**Mahmoud shibli, ID: 318539590**

**Lecturer: Dr. Sharon Yalov Handzel**

**Diabetes Prediction**

**Exploratory Data Analysis**

Authors: Saja mosa , Mahmoud shibli [Afeka*- Tel Aviv Academic College of Engineering]*

Keywords: Diabetes, Prediction, Healthcare, EDA, Deep learning, Machine learning, neural network

# **Abstract**

Diabetes is a chronic disease that requires early and accurate diagnosis for effective management. This study explores the impact of data preprocessing and neural network architecture modifications on diabetes prediction performance. We employ a deep learning model trained on the Diabetes Simple Diagnosis dataset from Kaggle, evaluating different data modification strategies. First, we improve model performance by removing outliers and misleading records, significantly enhancing accuracy (0.98) and recall (0.92). Conversely, modifying the dataset to introduce bias results in a notable decline in performance, with accuracy dropping to 0.76 and recall to 0.33. Additionally, enhancements to the neural network architecture, including increased layers, batch normalization, dropout regularization, and the application of SMOTE for class balancing, further optimize classification outcomes. Our findings underscore the importance of data preprocessing and network optimization in improving predictive accuracy, reducing false negatives, and ensuring reliable diabetes diagnosis.

# **Introduction**

Diabetes is a chronic metabolic disorder characterized by high blood sugar levels due to insufficient insulin production or the body's resistance to insulin. It primarily includes Type 1 and Type 2 diabetes, each with distinct causes and risk factors. The global prevalence of diabetes has risen dramatically, leading to significant public health challenges, including increased rates of morbidity and mortality. Type 1 diabetes is a chronic autoimmune condition in which the body's immune system mistakenly attacks the insulin-producing beta cells in the pancreas, resulting in little to no insulin production. The exact cause remains unclear, though genetic and environmental factors, such as viral infections, are believed to play a role. On the other hand, Type 2 diabetes, which accounts for about 90% of cases, develops due to insulin resistance, where the body's cells fail to respond properly to insulin, combined with a gradual decline in insulin production. This form of diabetes is closely linked to lifestyle factors such as obesity, physical inactivity, and poor diet, along with genetic predisposition. As of 2024, currently over  350 million adults worldwide (10.5% of the population aged 20-79) are living with diabetes. Projections indicate this number could rise to 783 million by 2045 Notably, over half of those affected are unaware of their condition. Early diagnosis is crucial to prevent severe complications such as cardiovascular diseases, kidney failure, nerve damage, and vision loss [[1]](#r1) [[2]](#r2).

# **Literature review**

In our analysis, we used a **diabetes classification dataset**, where previous studies have applied various machine learning models to assess their effectiveness. These studies tested models such as **Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression**, evaluating their performance based on accuracy scores. In these studies, **KNN initially achieved the highest accuracy of 95.5%,** while Decision Tree and Logistic Regression had slightly lower results. Further experiments showed variations, with **Decision Tree reaching the highest accuracy of 96.1%**, followed by **KNN with 95.6%,** and **Logistic Regression with 95.3%** [**[3]**](#r3) **.**

# **Methodology**

## **Data Presentation**

The dataset columns are shown in ([Table 1](#table_1)), including the dependent variable (Diagnosis) and the independent variables (all the other variables). Also, type and description were added for each variable.

**Table 1:** Variables Description

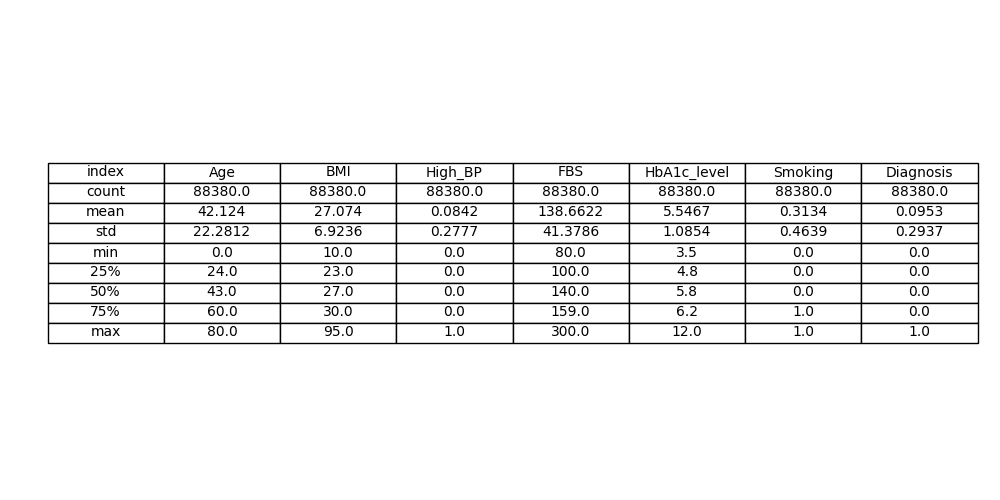
|  |  |  |
| --- | --- | --- |
| Variable | Type | Description |
| Age | Numeric, Integer | Patient age in years. Age can be a risk factor for diabetes, as the risk of diabetes increases with age. |
| Gender | Nominal, Binary | Indicates the gender of the patient, which can be a factor in diabetes prediction (Male/Female) |
| BMI | Numeric, Float | a measure that uses a person's height and weight to determine whether they are normal weight, overweight, or obese. |
| High\_BP | Binary, Integer | An indicator of whether the patient is suffering from high blood pressure or not (1: Yes/0:No) |
| FBS | Numeric, Integer | blood glucose level after overnight fasting (mg/dL) |
| HbA1c\_level | Numeric, Float | measurement of average blood sugar levels over the past 2-3 months |
| Smoking | Binary, Integer | Indicates whether the patient smokes or not (1: Yes / 0: No) |
| Diagnosis | Binary, Integer | An indicator that someone has diabetes (1: Yes/ 0:No) |

**checking null values in the dataset:** We recognize that there are no “null” values over all the data.

**Variables Transformation:** We performed Label encoding for the categorial variable “Gender”, by converting categorical data into a numeric format that machine/Deep learning models can process. It ensures consistency, efficiency, and proper model interpretation. This preprocessing step is essential for building effective and reliable models.

## **Analyzing Data**

Statistical values for all the variables displayed in ([Fig. 1](#fig1)), we have 88,380 records, including the mean, standard deviation, minimum, maximum, and percentiles for each variable. Variables with a high standard deviation indicate a large spread in values. For example, Age has a mean of 42.1 and a standard deviation of 22.28, suggesting a wide range from 0 to 80 years. Similarly, Fasting Blood Sugar (FBS) ranges from 80 to 300, with a mean of 138.66 and a standard deviation of 41.38, highlighting significant differences among individuals. Body Mass Index (BMI) also shows high variability, with a mean of 27.07 and a standard deviation of 6.92, spanning from 10 to 95. The high standard deviation in these variables indicates considerable dispersion in the data, meaning some individuals have very low values while others have very high ones.

****

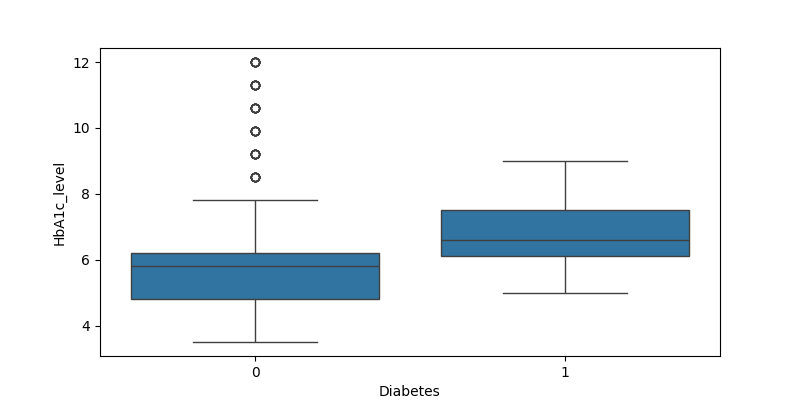
**Fig. 1.** Statistics summary for the dataset.

Visualization plots were built to understand the correlation and the relationship between diabetes and other variables. As displayed in ([Fig. 2](#fig2)), the count of individuals who reported suffering (1) not suffering (0) from diabetes, the majority of the individuals (**79956 individuals)** 90.47% have diabetes and (**8424 individuals)** 9.53% don't have diabetes, which means that we have an imbalanced dataset.

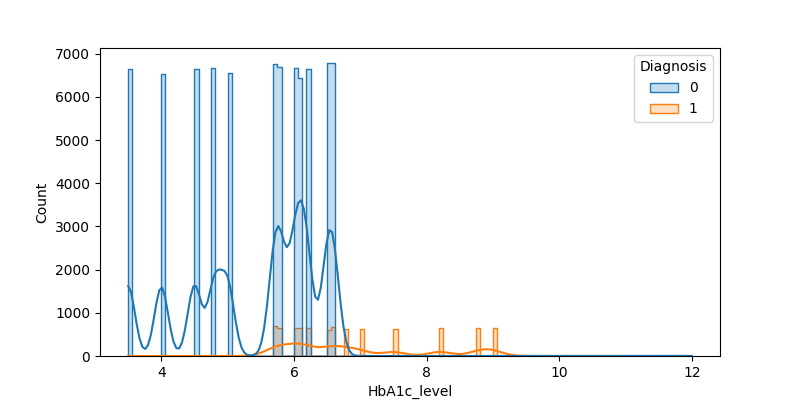
A graph of a diabetes status

AI-generated content may be incorrect.  
**Fig. 2.** Individuals count and percentage by Diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

The plots ([Fig. 3](#fig3)) and ([Fig. 4](#fig4)) show the distribution of HbA1c levels according to diabetes status. In the boxplot ([Fig. 3](#fig3)), it is evident that individuals with diabetes (1) tend to have higher HbA1c levels, while non-diabetic individuals (0) are concentrated around lower values. In the boxplot of non-diabetics, there are outliers (points outside the box), representing individuals with unusually high HbA1c levels despite not being diagnosed with diabetes. In contrast, there are almost no outliers among diabetics, as most of them have consistently high HbA1c values. In the distribution plot ([Fig. 4](#fig4)), among individuals with low HbA1c levels (around 4-6), there are almost no diabetics (1), meaning that the vast majority of people with these values are non-diabetic. On the other hand, as HbA1c levels increase, especially above 6.5, the number of diabetics rises significantly.

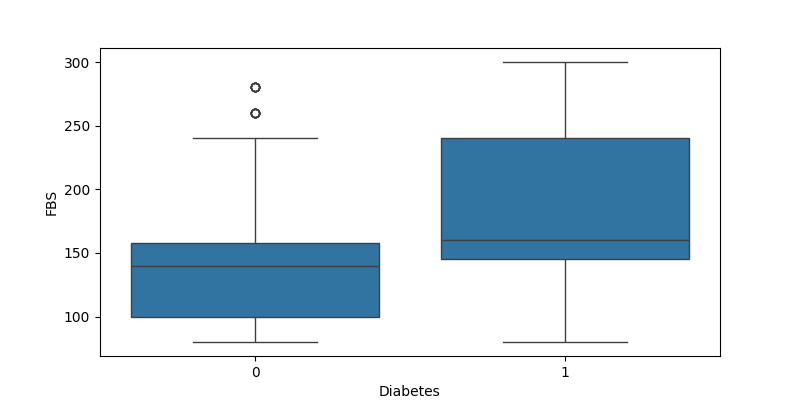


**Fig. 3.** Boxplot of HbA1c\_level by Diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

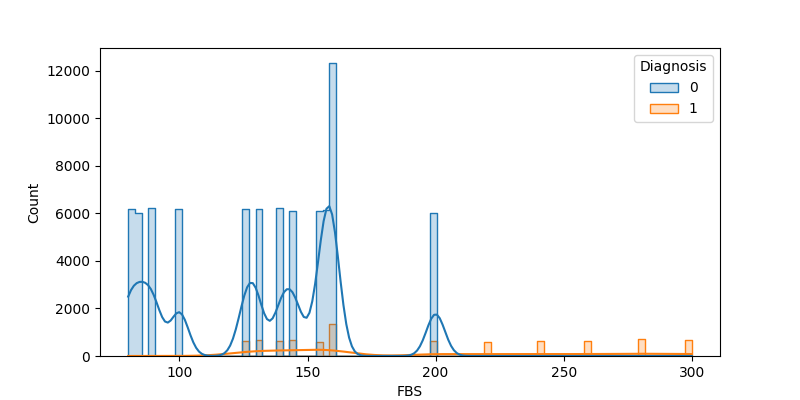


**Fig. 4.** HbA1c\_level distribution by diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

The plots ([Fig. 5](#fig5)) and ([Fig. 6](#fig6)) show the distribution of Fasting Blood Sugar (FBS) levels according to diabetes status. In the boxplot ([Fig. 5](#fig5)), it is evident that individuals with diabetes (1) tend to have higher FBS values compared to non-diabetics (0). The median and interquartile range for diabetics are significantly higher, indicating elevated blood sugar levels. Additionally, among non-diabetics, there are outliers (points above the typical range), representing individuals with unusually high FBS values despite not being diagnosed with diabetes. In contrast, among diabetics, there are almost no outliers, as most of them have consistently high FBS levels. In the distribution plot ([Fig. 6](#fig6)), most non-diabetic individuals are concentrated around FBS values of 100-150, while diabetics tend to have higher values, with a broader spread reaching up to 300. It is also noticeable that the number of diabetics is relatively low, but they appear almost exclusively at higher FBS levels.

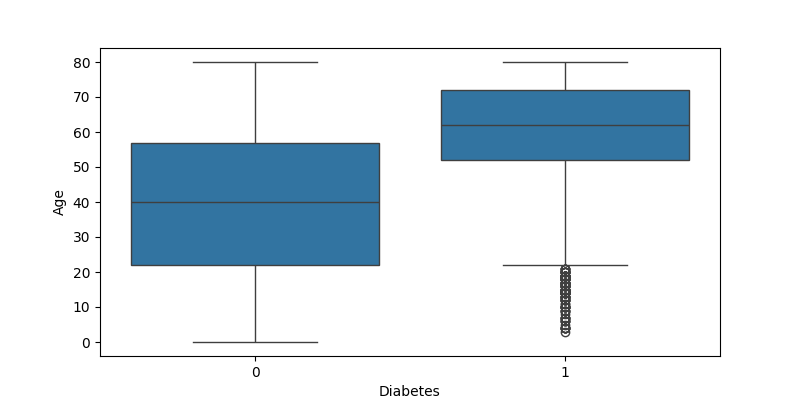


**Fig. 5.** Boxplot of FBS by Diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

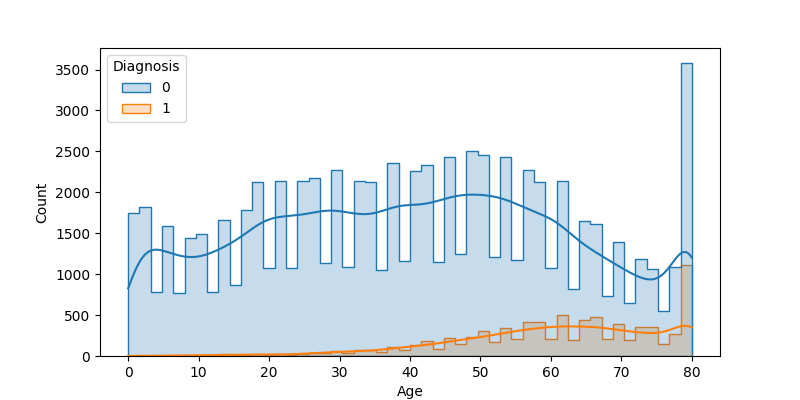


**Fig. 6.** FBS distribution by diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

The plots ([Fig. 7](#fig7)) and ([Fig. 8](#fig8)) present the age distribution according to diabetes status. ([Fig.7](#fig7)) is a boxplot that shows the age distribution for individuals with and without diabetes. It can be observed that the median age of the diabetic group is higher than that of the non-diabetic group, indicating that people with diabetes tend to be older. Additionally, several outliers can be identified at younger ages in the diabetic group, suggesting the presence of some cases of diabetes at a very young age. ([Fig. 8](#fig8)) displays the age distribution as a histogram divided by diabetes status. It shows that the non-diabetic population is spread across a wide age range, with a higher concentration at younger and middle-aged groups. In contrast, people with diabetes are mainly found at older ages, with their numbers increasing with age. This trend aligns with ([Fig.7](#fig7)), where diabetes is less common at younger ages, but some exceptional cases appear at very young ages.

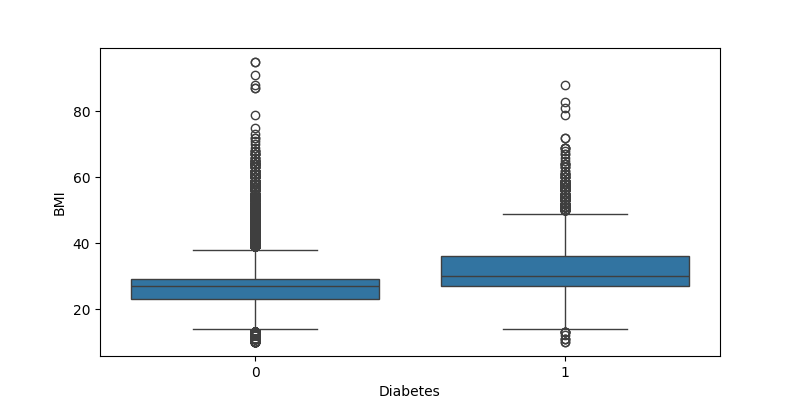


**Fig. 7.** Boxplot of Age by Diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

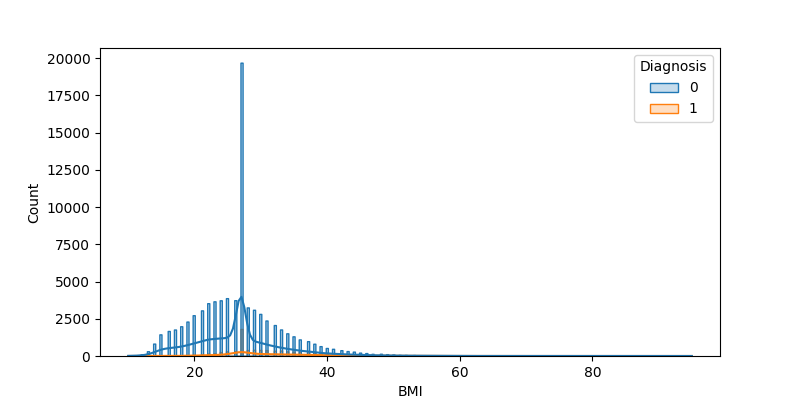


**Fig. 8.** Age distribution by diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

The plots illustrate the distribution of Body Mass Index (BMI) according to diabetes status. ([Fig. 9](#fig9)) is a boxplot that displays the distribution of BMI values among individuals with and without diabetes. It shows that the median BMI is higher in the diabetic group compared to the non-diabetic group, suggesting a potential link between higher BMI and an increased risk of developing diabetes. Additionally, there are numerous outliers in both groups, particularly at high BMI values, indicating the presence of individuals with extreme obesity, as well as some lower-end outliers. ([Fig. 10](#fig10)) presents the BMI distribution as a histogram, categorized by diabetes status. The majority of the population falls within the normal weight to overweight BMI range, with a tendency towards higher BMI values. Among individuals with diabetes, there is a noticeably higher prevalence of elevated BMI, reinforcing the association between excess weight and diabetes risk. Furthermore, extreme BMI values can be observed, representing cases of severe obesity.

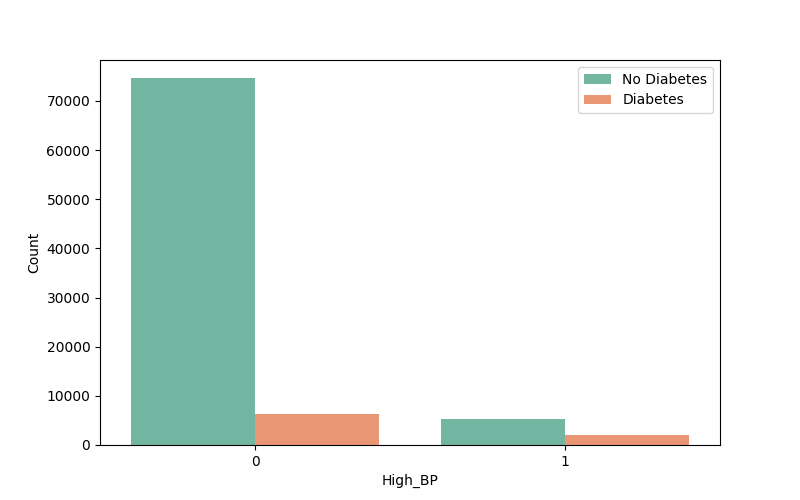


**Fig. 9.** Boxplot of BMI by Diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).



**Fig. 10.** BMI distribution by diabetes status, ‘1’= (with diabetes), ‘0’= (without diabetes).

The plot ([Fig. 11](#fig11)) displays the number of individuals with and without diabetes, grouped by high blood pressure (High\_BP). It shows that the majority of the population does not have high blood pressure. However, among individuals with diabetes, there is a higher proportion of those with high blood pressure compared to those without diabetes. This trend is also reflected in ([Table 2](#table_2)), which indicates that 24.81% of individuals with diabetes have high blood pressure, compared to only 6.69% among non-diabetic individuals. This suggests a potential association between diabetes and high blood pressure.

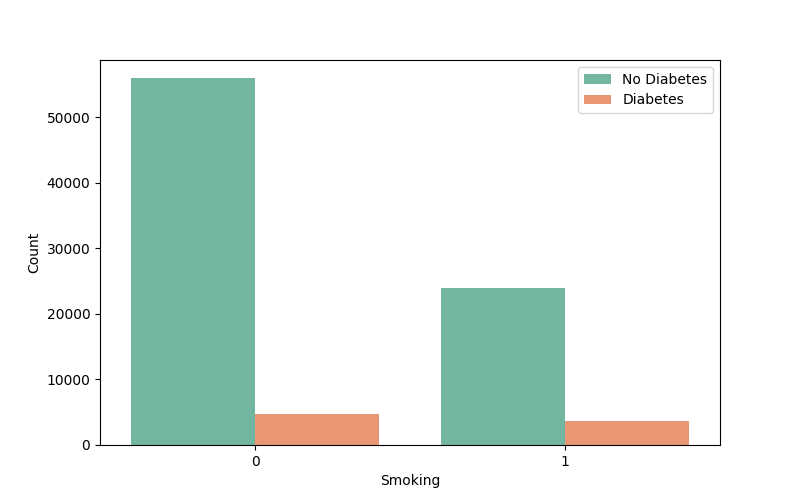


**Fig. 11.** Count of Individuals with and without Diabetes, Grouped by High\_BP, ‘1’= (with High\_BP), ‘0’= (without High\_BP).

**Table 2:** Distribution by the percentage of Diabetes and high\_BP Among Participants

|  |  |  |
| --- | --- | --- |
|  | Diabetes | Without Diabetes |
| high\_BP | 24.81% | 6.69% |
| Without high\_BP | 75.19% | 93.31% |

The plot in ([Fig. 12](#fig12)) presents the number of participants (y-axis) with and without diabetes, categorized by smoking status – 0 (non-smokers) and 1 (smokers) on the x-axis. It shows that most non-smokers do not have diabetes, while the proportion of individuals with diabetes is higher among smokers. This trend is also supported by ([Table 3](#table_3)) where the percentage of diabetics is higher among smokers (44.05%) compared to non-smokers (55.95%).

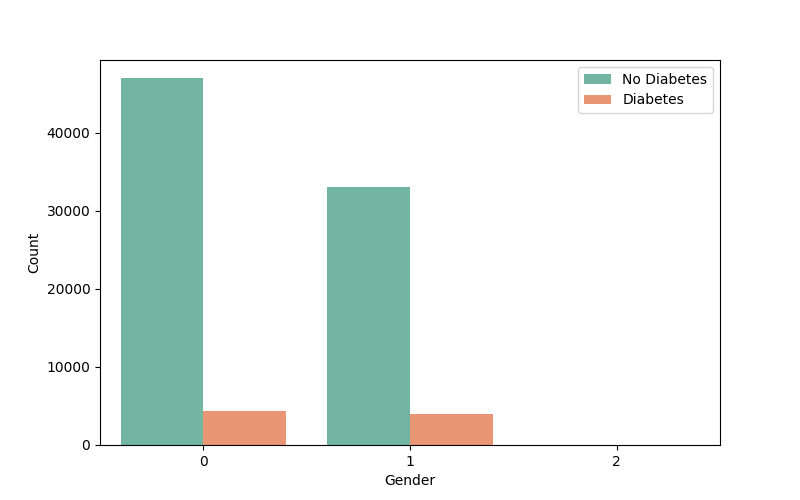
****

**Fig. 12.** Count of Individuals with and without Diabetes, Grouped by Smoking, ‘1’= (Smoking), ‘0’= (don’t smoke).

**Table 3:** Distribution by percentage of Diabetes and Smoking Among Participants

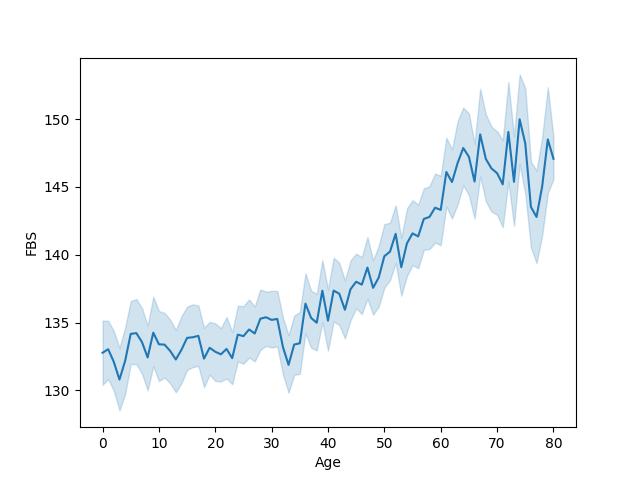
|  |  |  |
| --- | --- | --- |
|  | Diabetes | Without Diabetes |
| Smoking | 44.05% | 30.0% |
| Don’t Smoking | 55.95% | 70.0% |

The plot in ([Fig. 13](#fig13)) shows the number of participants (y-axis) with and without diabetes, categorized by gender – 0 (Female), 1 (Male), and 2 (Other) on the x-axis. Most participants are either male or female, with very few in the "Other" category. Additionally, most individuals in each gender group do not have diabetes, while the proportion of those with diabetes appears relatively similar between males and females.

****

**Fig. 13.** Count of Individuals with and without Diabetes, Grouped by Gender, ’2’= (Other), ‘1’= (Male), ‘0’= (Female).

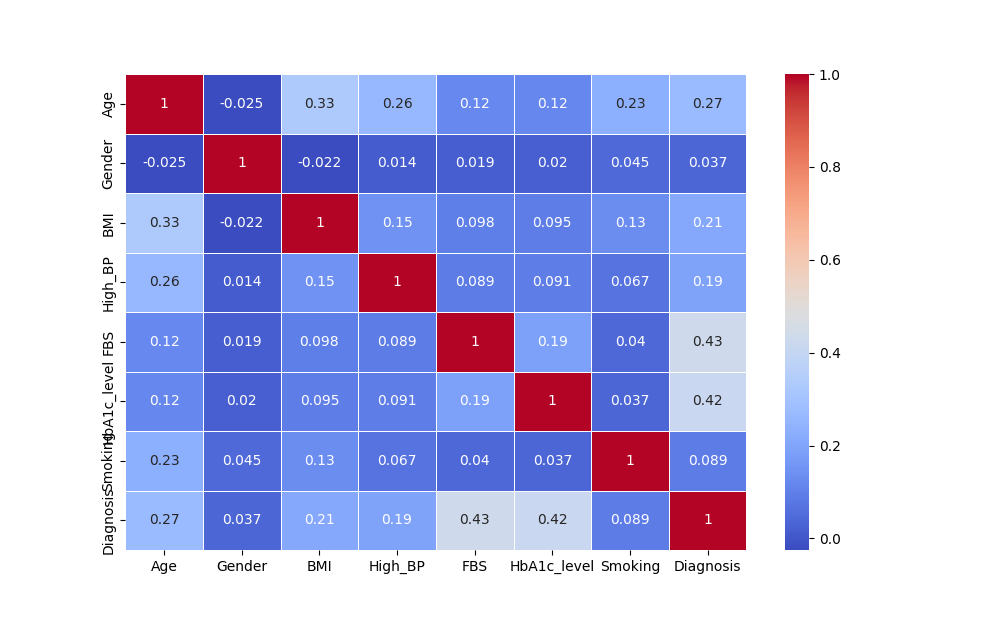
The plot shows fasting blood sugar (FBS) levels as a function of age ([Fig. 14](#fig14)). There is an upward trend, indicating that FBS levels tend to increase with age. The shaded area around the line represents the range of variability or uncertainty in the data. The increase in FBS levels becomes more noticeable after age 40, with a steeper rise in older ages.

****

**Fig. 14.** FBS by Age.

## **Correlation Matrix**

The correlation matrix in ([Fig. 15](#fig15)) shows the relationships between different features related to diabetes. HbA1c levels and FBS have the highest correlation with diabetes diagnosis (0.42 and 0.43, respectively), indicating that higher values of these variables are strongly associated with diabetes. Age and BMI also show a positive correlation (0.27 and 0.23) with diabetes diagnosis, though to a lesser extent. Smoking and high blood pressure (High\_BP) have a very weak correlation with diabetes diagnosis.



**Fig. 15.** Correlation Matrix of Diabetes Dataset Features

## **Dataset Preparation**

**normalization**: we're applying a technique called **normalization** to scale our numerical features, many machine/deep learning algorithms, especially those involving distance calculations or gradient descent, perform better when features are on a similar scale. If we have features with vastly different ranges (e.g., fasting blood sugar ranging from 80 to 300), the feature with the larger range might dominate the learning process, while the feature with the smaller range might be effectively ignored.

**Split into training, validation, and test sets**: we're splitting our dataset into three subsets **(training, validation, and test). By splitting our data, we can train a model that not only performs well on the data it was trained on but also generalizes effectively to new, unseen data, which is the ultimate goal of machine learning. Initial split:** We first split the data into 70% for training and 30% for temporary holding, **Second split:** we then split the temporary data into 50% for validation and 50% for testing. This results in approximately 15% of the original data for validation and 15% for testing.

## **Decision Tree Model**

We decided to use the decision tree model, in this model, we will be predicting if the individual has diabetes based on their medical and demographic records. **Decision Tree Classifier** is a popular and interpretable machine learning algorithm. We will use this as a baseline model before exploring more complex models like neural networks. Decision Trees are relatively easy to understand and visualize. They make predictions by constructing a tree-like structure where each node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome (in this case, diabetes or no diabetes). To assess the effectiveness of our Decision Tree model, we'll use a variety of metrics that provide a comprehensive view of its performance. **Accuracy**: this will give us an overall sense of how often the model predicts correctly. **Precision**: will tell us how reliable a positive prediction is, which is crucial in a medical context. **Recall**: on the other hand, will reveal how well the model identifies actual positive cases, ensuring we don't miss potential diagnoses. **F1 score**: will provide a balanced measure by combining precision and recall. To understand the model's ability to discriminate between classes, we'll examine the **ROC AUC**. Finally, the **confusion matrix:** will give us a detailed breakdown of the model's predictions, highlighting where it excels and where it might struggle, such as misclassifying true positives or negatives.

## **Neural Network Model**

We're building, training, and evaluating a **neural network model** for diabetes diagnosis. This allows us to explore a more complex model and compare its performance with the simpler Decision Tree. We are creating a sequential neural network model, the model has three layers: A dense layer with 64 neurons and ReLU activation: This is the first hidden layer, it takes the input features and applies a linear transformation followed by the ReLU activation function, which introduces non-linearity. Dense layer with 32 neurons and ReLU activation: This is the second hidden layer, further processing the information from the first layer. Dense layer with 1 neuron and sigmoid activation: This is the output layer, the sigmoid activation function produces a probability between 0 and 1, representing the likelihood of diabetes.

We configured the learning process with the Adam optimizer which is an efficient algorithm for training neural networks, the learning rate is default value (0.001), utilizing loss function: binary cross-entropy to gauge the disparity between predictions and actual diagnoses, and monitoring accuracy throughout training. The network was then trained on our data, iterating over it 50 times (epochs= 30), with performance validated after each epoch. The model updates its weights after processing 32 samples at a time (batch\_size= 32), then we used the trained neural network to make predictions on the test set. The predictions are converted to 0 or 1 based on a threshold of 0.5. We evaluate the neural network's predictions using the same metrics as the Decision Tree.

## **Hyperparameters**

### **Neural number in the hidden layer**

we performed hyperparameter tuning by varying the number of neurons (units) in the first hidden layer of a neural network. We tested three different values for the number of neurons in the first hidden layer (128, 256, 1024) and evaluated how they impact model performance, other inputs were still constant (epochs= 30, learning rate= 0.001, batch size= 32). This helps in finding the optimal number of neurons to achieve better accuracy, precision, recall, F1-score, and ROC AUC.

### **Learning rate**

**We choose hyperparameter tuning** by varying the **learning rate** of the optimizer and evaluating the model's performance. The model was tested by three values of the learning rate (0.0001, 0.0008, 0.1). Other inputs are still constant (epochs= 30, Neural number in hidden layer= (64, 32,1), batch size= 32). The model was evaluated by the following metrics: accuracy, precision, recall, F1-score, and ROC AUC.

### **Batch size**

**hyperparameter tuning** by varying the **batch size** to observe how it impacts model performance. The **batch size** determines how many training samples the model processes before updating the weights. We chose different batch sizes (10, 100, and 500) to analyze their impact on model performance. Other inputs are still constant (epochs= 30, Neural number in hidden layer= (64, 32,1), learning rate= 0.001). By testing these values of batch size, we can determine which batch size yields the best accuracy, precision, recall, F1-score, and ROC AUC for my model.

## **Modify the dataset to improve the results improve significantly**

We deleted records from the data to significantly improve the evaluation metrics. We used a dataset analysis that was performed. After analyzing the data, we concluded that if we delete the rows that have outliers, that do not have diabetes, and that there is another variable that causes diabetes, then we will delete these rows. We decided to delete records that do not have diabetes and that have HbA1c\_level >6, look at ([Fig. 3](#fig3)) that there are outliers for these records. We also deleted records that do not have diabetes and have an FBS>150 based on outliers in ([Fig. 5](#fig5)), we find the normalized value that is equal to the original values and delete those records from the new db we want to train .By deleting records, we reduce the chance of False negatives and expand the chance of True positives.

## **Modify the dataset to make the results worse significantly**

We deleted records from the data to get significantly worse evaluation metrics. We deleted records for the numerical variables, we approximated their range between subjects who have diabetes and those who do not. We decided to delete records that (1) do not have diabetes and have HbA1c\_level <5, (2) have diabetes and have HbA1c\_level >7, look at ([Fig. 3](#fig3)). (3) have diabetes and have FBS >150, look at ([Fig. 5](#fig5)). (4) have diabetes and have Age <50, (5) don’t have diabetes and have Age <50 look at ([Fig. 7](#fig7)). (6) have diabetes and have BMI <40, (7) don’t have diabetes and have BMI <40 look at ([Fig. 9](#fig9)).

## **improve the network architecture**

we made several improvements to your neural network architecture and the training process, to improve the training, **SMOTE (Synthetic Minority Over-sampling Technique)** applied to oversample the minority class in the training data, we use bigger batch size 500. to **improve the neural network architecture** we added three dense layers with 128, 64, 32 units and dded batch normalization after each dense layer, additionally a dropout rate of 0.5 is used, the output layer has a single neuron with a sigmoid activation function. The learning rate reduced to **0.000.**

## **New evaluation metric**

We choose Learning Efficiency Ratio (LER) for 2F1 metric that used to assess how effectively a model is learning during training. The ratio indicates the change in 2-F1 score per epoch. A positive value means the model is improving (2-F1 score is increasing), while a negative value means the model is performing worse (2-F1 score is decreasing).

## **data balance changing**

we explored the impact of different levels of class imbalance on a neural network's performance by modifying the distribution of the target variable (Diagnosis) in the training dataset. To create different imbalance levels, we applied **oversampling and undersampling techniques. Balanced Data (50:50)** – **minority: majority classes,** we used **SMOTE (Synthetic Minority Over-sampling Technique)**. **Moderate Imbalance (70:30)** –– **minority: majority classes,** **random undersampling** applied. **Severe Imbalance (90:10)** – **minority: majority classes.** For each of these three datasets, we trained a simple **neural network classifier** with two hidden layers (64 and 32 neurons, both using ReLU activation). The model was compiled with the Adam optimizer and binary cross-entropy loss. Training was conducted for **30 epochs**, and the model's performance was evaluated using the **validation accuracy** metric.

## **dimensionality reduction method**

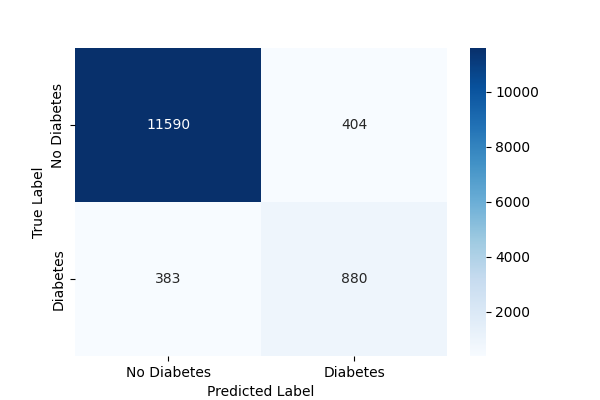
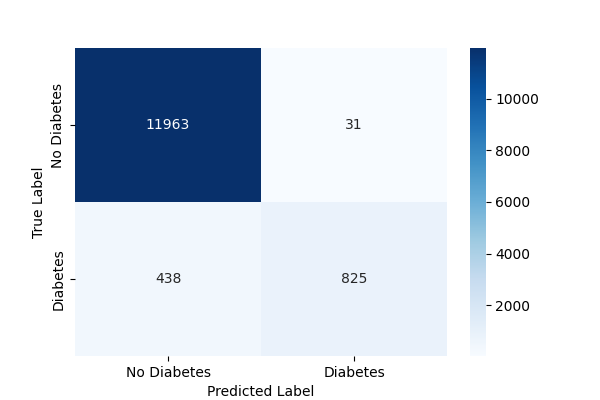
**We performed Principal Component Analysis (PCA) to reduce the dimensionality of your dataset while retaining most of the variance. Visualizes the dataset in 2D to see how well the data is separated. Trains a neural network on the reduced dataset and evaluates performance.**

# **Results**

We can see evaluation metrics for two algorithms ([Table 4](#table_4)): Decision Tree and Neural Network. It shows that the Neural Network has a higher accuracy compared to the Decision Tree. The Precision value is higher for the Neural Network, while the Recall value is higher for the Decision Tree. The F1 Score is also higher for the Neural Network, whereas the ROC AUC value is the same for both models. Confusion matrix for both models displayed in ([Fig. 16](#fig16)), a large number of samples were correctly classified as "No Diabetes" in both models, we calculated the evaluation metrics based on the confusion matrix.

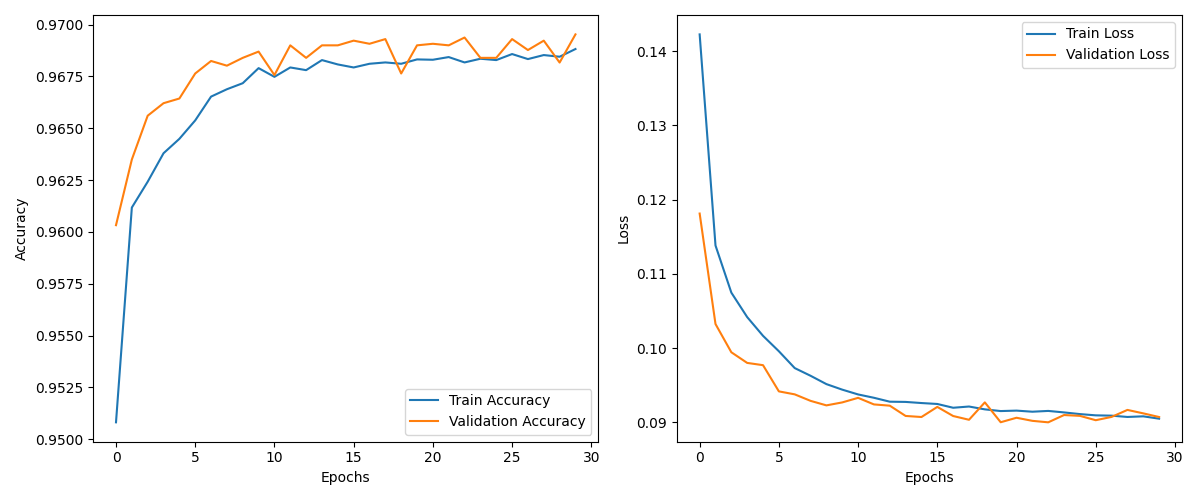
**Table 4:** evaluation metrics in decision tree and neural network methods

|  |  |  |
| --- | --- | --- |
| **Evaluation metrics** | **Decision Tree** | **Neural Network** |
| **Accuracy** | 0.94 | 0.96 |
| **Precision** | 0.69 | 0.96 |
| **Recall** | 0.70 | 0.65 |
| **F1 Score** | 0.69 | 0.78 |
| **ROC AUC** | 0.83 | 0.83 |

**** 

**Fig. 16.** Confusion matrix for the decision tree (left plot), neural network (right plot)

In the left plot ([Fig. 17](#fig17)), the training and validation accuracy increases rapidly at the beginning and stabilize around epochs 10-15, with the validation accuracy fluctuating slightly around the training accuracy afterward. In the right plot ([Fig. 17](#fig17)), the training and validation loss decrease quickly during the initial epochs and stabilize around epochs 15-20, with minor variations and Volatility afterward.

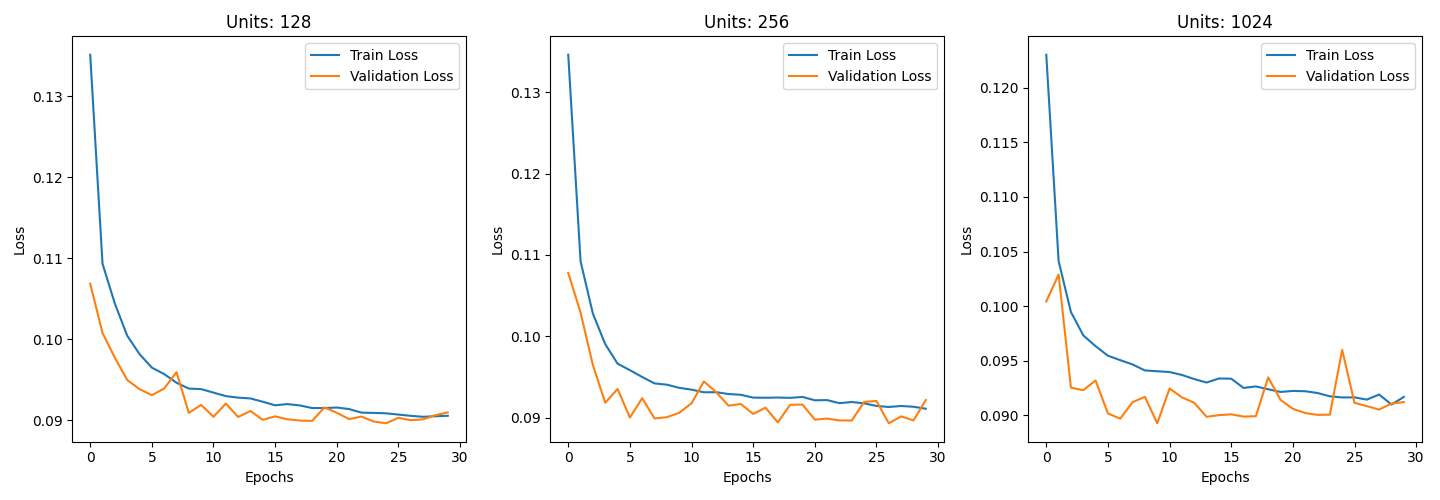


**Fig. 17.** Training and Validation Accuracy Over Epochs (left plot), Loss Over Epochs (right plot) for neural network.

In ([Table 5](#table_5)), accuracy is slightly higher when the number of neurons increases. Precision also improves as the number of neurons increases. Recall remains similar, with a slight decrease when the number of neurons is very high. The F1 Score and the ROC AUC value stay the same across all cases. In ([Fig. 18](#fig18)), three plots show the training and validation loss curves over 30 epochs for neural networks with different hidden layer sizes: 128, 256, and 1024 units. All three models show a sharp decrease in loss within the first few epochs. The 128 and 256-unit models have relatively stable validation loss with minor fluctuations. The 1024-unit model has higher fluctuations in validation loss, suggesting potential overfitting.

**Table 5:** evaluation metrics in neural network methods with different Neural numbers in the hidden layer.

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation metrics** | **Neural number = 128** | **Neural number = 256** | **Neural number = 1024** |
| **Accuracy** | 0.96 | 0.97 | 0.97 |
| **Precision** | 0.97 | 0.98 | 0.99 |
| **Recall** | 0.65 | 0.65 | 0.64 |
| **F1 Score** | 0.78 | 0.78 | 0.78 |
| **ROC AUC** | 0.82 | 0.82 | 0.82 |

****

**Fig. 18.** Training and Validation Loss for different units

In ([Table 6](#table_6)), the accuracy is over 0.9 at all the learning rate values, the other evaluation metrics are the highest at a moderate learning rate (0.005), and there are close values for the learning rate () to the moderate learning rate and drops significantly at a high learning rate (0.5). The plots in ([Fig. 19](#fig19)), demonstrate the impact of different learning rates on model training. A low learning rate (LR = 0.0001) results in slow but steady convergence. Increasing the learning rate to 0.005 accelerates learning but introduces instability. A high learning rate (LR = 0.1) leads to rapid initial progress but ultimately prevents the model from finding a good solution.

**Table 6:** evaluation metrics in neural network methods with different learning rate

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation metrics** | **Learning rate =** | **Learning rate = 0.005** | **Learning rate = 0.5** |
| **Accuracy** | 0.96 | 0.97 | 0.90 |
| **Precision** | 0.95 | 0.99 | 0.00 |
| **Recall** | 0.62 | 0.64 | 0.00 |
| **F1 Score** | 0.75 | 0.78 | 0.00 |
| **ROC AUC** | 0.81 | 0.82 | 0.5 |

A graph with blue and orange lines

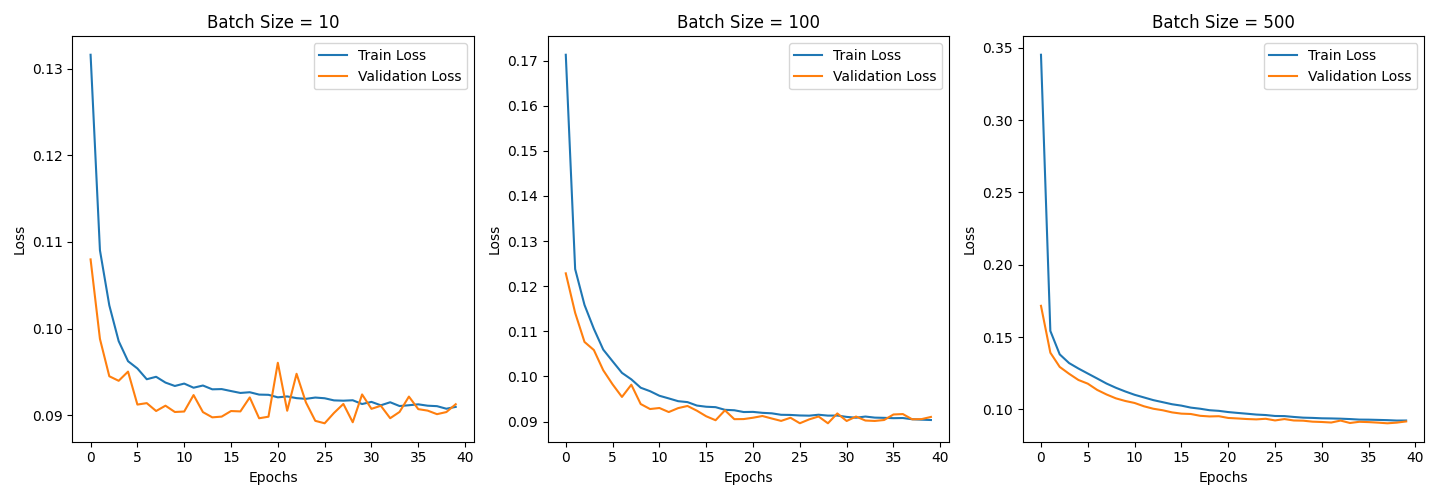
AI-generated content may be incorrect.

**Fig. 19.** Training and Validation Loss for different learning rates

In ([Table 7](#table_7)), accuracy remains the same across different batch sizes. Precision is highest for the smallest batch size and decreases slightly as batch size increases. Recall remains similar across all batch sizes, with a slight increase at the largest batch size. The F1 Score stays the same for all batch sizes. The ROC AUC value is slightly higher for larger batch sizes. In the ([Fig. 20](#fig20)), the training loss starts high and decreases over time for all batch sizes, stabilizing around epoch 30-40. Larger batch sizes show a smoother loss curve with less fluctuation, while smaller batch sizes have more variance in validation loss.

**Table 7:** evaluation metrics in neural network methods with different batch size

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation metrics** | **Batch size = 10** | **Batch size = 100** | **Batch size = 500** |
| **Accuracy** | 0.97 | 0.97 | 0.97 |
| **Precision** | 0.99 | 0.97 | 0.96 |
| **Recall** | 0.65 | 0.65 | 0.66 |
| **F1 Score** | 0.78 | 0.78 | 0.78 |
| **ROC AUC** | 0.82 | 0.83 | 0.83 |



**Fig. 20.** Training and Validation Loss for different batch sizes

We can see in ([Table 8](#table_8)) that accuracy is highest when the neural network is modified for improvement and lowest when modified for worse. Precision increases with improvement and decreases significantly when modified for worse. Recall is much higher with improvement and drops sharply when modified for worse. The F1 Score, ROC AUC values follow the same pattern, being highest when modified for better and lowest when modified for worse.

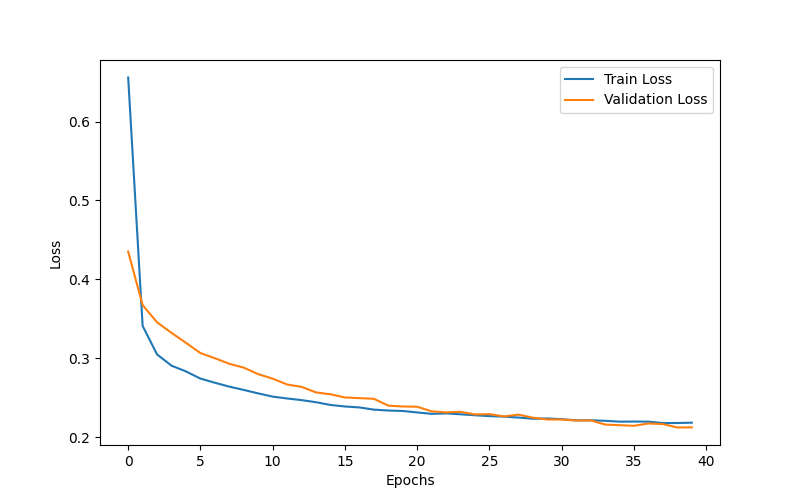
**Table 8: Comparis**on of theevaluation metrics between the original neural network with modify to better and modified to worse.

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation metrics** | **Original neural network** | **modify to better** | **Modify to worse** |
| **Accuracy** | 0.96 | 0.98 | 0.76 |
| **Precision** | 0.96 | 0.99 | 0.59 |
| **Recall** | 0.65 | 0.92 | 0.33 |
| **F1 Score** | 0.78 | 0.96 | 0.42 |
| **ROC AUC** | 0.83 | 0.96 | 0.62 |

In ([Table 9](#table_9)), accuracy decreases in the improved architecture compared to the original neural network. Precision is significantly lower in the improved architecture, while recall is much higher. The F1 Score is lower in the improved architecture. The ROC AUC value is higher in the improved architecture. In the graph ([Fig. 21](#fig21)), training and validation loss both decrease over epochs, with the validation loss closely following the training loss, showing a smooth decline over time.

**Table 9:** Comparison of theevaluation metrics between the original neural network and the improved architecture.

|  |  |  |
| --- | --- | --- |
| **Evaluation metrics** | **Original neural network** | **Improved architecture** |
| **Accuracy** | 0.96 | 0.88 |
| **Precision** | 0.96 | 0.45 |
| **Recall** | 0.65 | 0.94 |
| **F1 Score** | 0.78 | 0.60 |
| **ROC AUC** | 0.83 | 0.91 |



**Fig. 21.** Training and Validation Loss after improved architecture

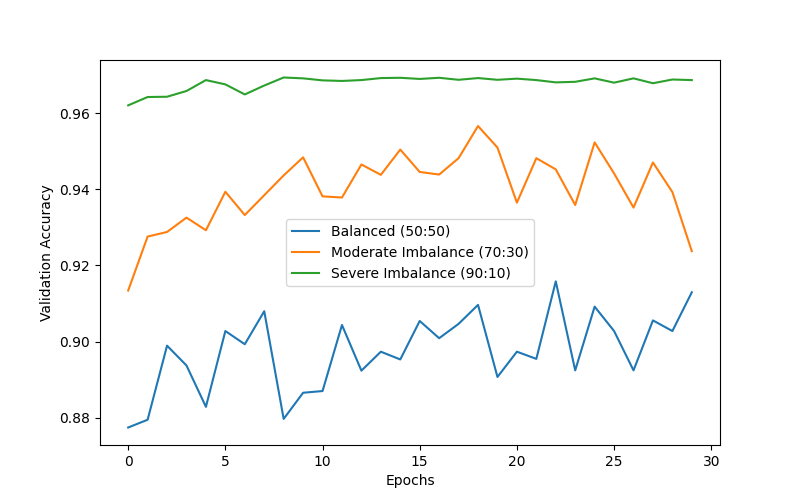
The plot in ([Fig. 22](#fig22)) suggests that the model learned effectively in the initial epochs (Epochs 1-5), albeit with some instability. After the initial learning phase, the model converged and its performance stabilized (Epochs 6-30). From around epoch 20 onwards, the LER-2F1 is very close to zero.

A graph with a line

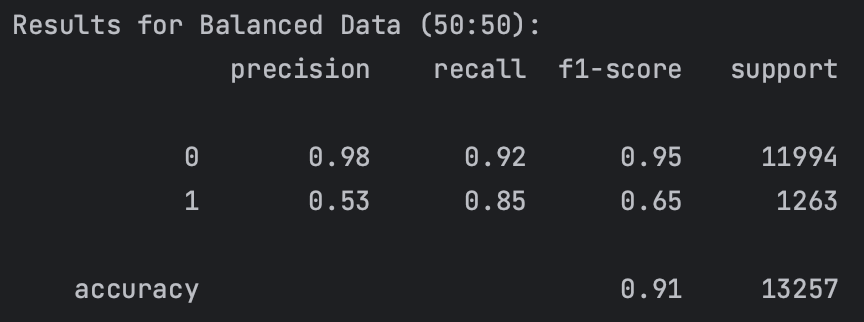
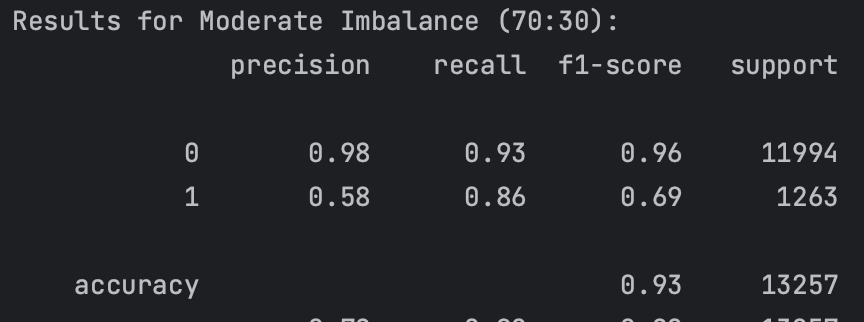
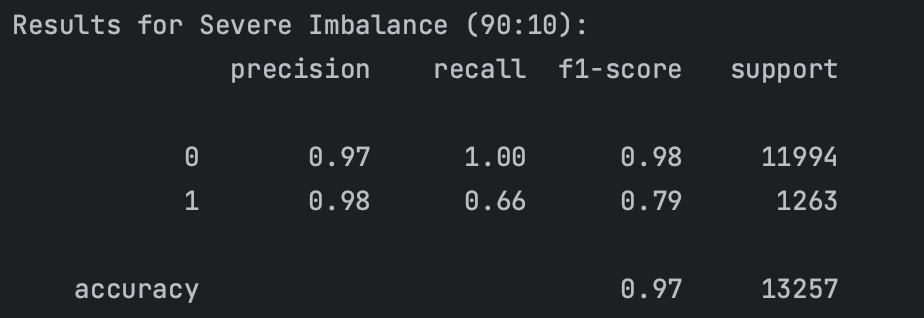
AI-generated content may be incorrect.

**Fig. 22.** Training and Validation for 2F1-LER over epochs

The plot in ([Fig. 23](#fig23)) displays model performance across different class imbalances over multiple epochs. The three lines represent balanced data (50:50), moderate imbalance (70:30), and severe imbalance (90:10). The balanced data maintains the highest and most stable accuracy throughout training, while the moderate imbalance shows slightly lower accuracy with some fluctuations. The severe imbalance demonstrates the most instability, with noticeable fluctuations in performance over epochs. The ([Fig. 24](#fig24)) displays evaluation metrics for different class imbalance scenarios. In the severe imbalance case (90:10), class 0 has high precision and recall, while class 1 exhibits lower recall, leading to a high overall accuracy. In the moderate imbalance case (70:30), the recall for class 1 improves compared to the severe imbalance scenario, but the precision for class 0 decreases slightly. In the balanced data case (50:50), the recall for class 1 reaches its highest value, but precision decreases, and overall accuracy is slightly lower than in the imbalanced cases.

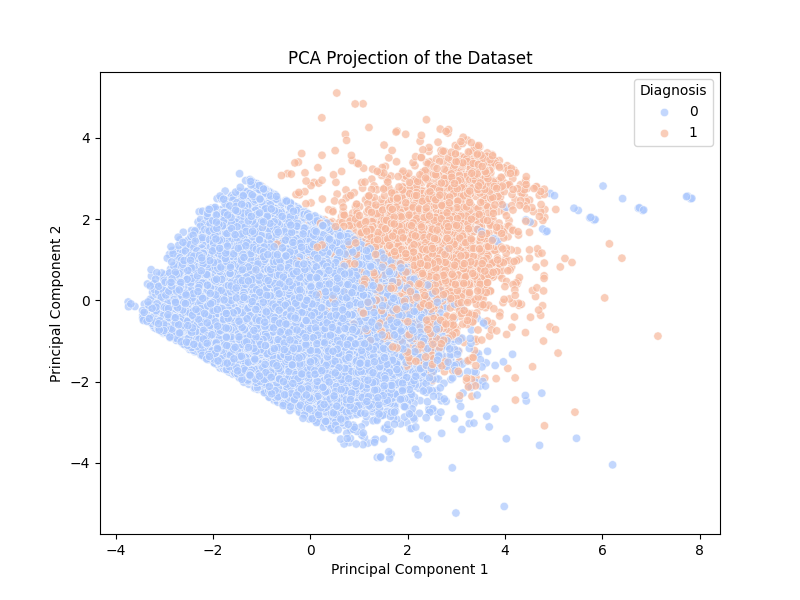


**Fig. 23.** Model Performance with Different Class Imbalances

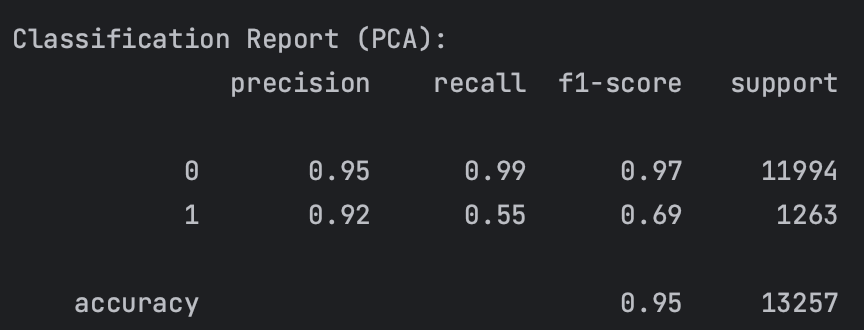


**Fig. 24.** Evaluation metrics for Different Class Imbalances

The PCA plot ([Fig. 25](#fig25)) shows how well the data points representing "no diabetes" (diagnosis 0) and "diabetes" (diagnosis 1) are separated when projected onto the first two principal components. The partial separation we observed in the plot suggests that there are underlying patterns in the data that can help distinguish between these two diagnoses. However, the overlap indicates that the separation is not perfect. In the classification report ([Fig. 26](#fig26)), The model performs very well in identifying individuals without diabetes. **Precision:** it's correct 95% of the time. **Recall:** 99% of all individuals who actually don't have diabetes. **F1-score (0.97):** This high F1-score indicates a good balance between precision and recall. The model's performance in identifying individuals with diabetes is less strong, **Precision:** When the model predicts "diabetes," it's correct 92% of the time. **Recall:** the model only identifies 55% of all individuals who actually have diabetes. **F1-score (0.69):** The lower recall significantly impacts the F1-score. The overall accuracy 95%.



**Fig. 25.** PCA Projection of the Dataset



**Fig. 26.** Evaluation metrics for PCA method

# **Discussion and Conclusions**

The evaluation metrics ([Table 4](#table_4)) indicate that the neural network outperforms the decision tree in accuracy, precision, and F1 score, indicating its better overall performance. While both models have the same ROC AUC. The Neural Network's higher precision implies fewer false positives. However, the Decision Tree exhibits slightly better recall, meaning it captures more true positive cases. Overall, the neural network demonstrates stronger performance, particularly in reducing false positives, making it a preferable choice for this classification task. The neural network ([Fig. 17](#fig17)) shows a rapid increase in both training and validation accuracy in the initial epochs, suggesting the model learns quickly but reaches a performance ceiling. The decrease in both training and validation loss, indicates effective learning and convergence. However, the validation loss shows slight fluctuations, hinting at potential minor overfitting or sensitivity to the validation data. the model demonstrates good learning behavior with rapid convergence and stable performance, but needs to improve generalization and prevent overfitting.

We performed hyperparameter tuning for three hyperparameters, To find the optimal hyperparameter value that gives the best model performance. **(1)** Regarding the results in ([Table 5](#table_5)) and ([Fig. 18](#fig18)), By comparing results across different values of units to understand how the **model capacity** affects accuracy and other metrics. All three models **converge to a similar validation loss (~0.09)**, suggesting that increasing the number of units does not drastically improve generalization. The 1024-unit model exhibits more variation in validation loss, suggesting it may be overfitting to the training data, Although the **128 or 256-unit models** are preferable due to their stability and efficiency, while the 1024-unit model may be unnecessarily complex. **(2)** according to results in ([Table 6](#table_6)) and ([Fig. 19](#fig19)), A low learning rate leads to slow learning, while a high learning rate can cause instability and prevent the model from finding an optimal solution. **LR = 0.005 seems promising but potentially needs refinement.** It offers a balance between convergence speed and stability, but the validation loss exhibits some fluctuations. the optimal learning rate likely lies somewhere between 0.0001 and 0.005. **(3)** results in ([Table 7](#table_7)) and ([Fig. 20](#fig20)) show that batch size impacts training stability and convergence speed. Larger batch sizes (500) offer smoother training but slower initial progress, while smaller batches (10) introduce more noise but can accelerate initial learning. Smaller batch sizes introduce more randomness, helping generalization, while larger batches tend to stabilize convergence but may get stuck in local minima. The optimal batch size depends on the specific dataset and model, balancing stability and efficiency.

We performed 2 separated modification for the dataset, deleting records to get results significantly better, and delete records to get worse results significantly, the results in (Table 8) shows that when we removed outliers and records that could contribute to false negatives, the[table\_8](#table_8) evaluation metrics improved significantly, with accuracy increasing from 0.96 to 0.98 and recall from 0.65 to 0.92, indicating a better ability to identify diabetic cases. This suggests that removing misleading data points enhances classification performance. Conversely, when records were deleted in a way that distorted the data distribution, the model's performance deteriorated, with accuracy dropping to 0.76 and recall to 0.33, leading to a higher false negative rate. These findings emphasize the importance of proper data preprocessing in achieving reliable and accurate predictions in diabetes classification. The results of improved architecture shown in ([Table 9](#table_9)) and ([Fig. 21](#fig21)), the smoot helps in balancing the dataset, allowing the model to learn better representations for both classes. by added three dense layers with 128, 64, and 32 units, respectively, these layers allow the model to learn more complex patterns from the data. adding batch normalization after each dense layer, which helps in stabilizing and speeding up the training process by normalizing the activations of the previous layer. A dropout rate of 0.5 is used, which helps prevent overfitting by randomly dropping half of the units in each layer during training and smaller learning rate helps avoid overshooting the optimal weights, especially when using a more complex model.

The plot in ([Fig. 22](#fig22)) shows significant fluctuations in the LER-2F1 during the first 5 epochs. This suggests that the model is making large adjustments to its weights in the early stages of training, leading to unstable performance. This is not uncommon in the initial phase as the model tries to find a good starting point. **Convergence (Epochs 6-30):** After the initial volatility, the LER-2F1 stabilizes and gradually approaches zero. This indicates that the model is converging and its performance is becoming more consistent. The fluctuations become smaller and less frequent, suggesting that the model is making smaller and more refined adjustments to its weights.  **(Epochs 20-30):** From around epoch 20 onwards, the LER-2F1 is very close to zero. This implies that the model's performance is no longer significantly improving or degrading. The model has likely reached a point where further training is unlikely to yield substantial gains. Based on the plateau observed after epoch 20, it might be beneficial to implement early stopping to prevent unnecessary training and potential overfitting. The ([Fig. 23](#fig23)) and ([Fig. 24](#fig24)) shown analyzes and evaluates performance metrics for model performance under different class imbalances, the results show that when data is balanced, the model achieves the highest performance. However, as class imbalance increases, accuracy becomes less stable, indicating that the model struggles to generalize when trained on highly imbalanced data. Under severe imbalance (90:10), the model maintains high accuracy but suffers from low recall for the minority class. In contrast, when data is balanced (50:50), recall improves significantly, though overall accuracy slightly decreases. This suggests that class imbalance leads to biased predictions favoring the majority class.

We perform PCA dimensionality reduction method, the results in ([Fig. 25](#fig25)) and ([Fig. 26](#fig26)), indicate some level of distinction between the two classes, but there is still considerable overlap, suggesting that the features used may not completely separate the two categories. that the model performs better at identifying non-diabetic patients than diabetic ones, leading to lower recall for the diabetic class. It may occurs because the majority of the data is for the non-diabetic patients.

# **Future work**

Future advancements in deep learning for diabetes diagnosis can focus on integrating additional data sources and leveraging advanced techniques to enhance prediction accuracy and model interpretability. Implementing multimodal learning, which combines blood test results, medical imaging (such as retinal fundus scans), and wearable sensor data (such as continuous glucose monitoring), can improve early diagnosis and enable personalized treatment recommendations. Additionally, federated learning can be employed to train models across multiple healthcare datasets while preserving patient privacy.

# **References**

[1] Diabetes. (2024).[diabetes](https://exonpublications.com/index.php/exon/issue/view/diabetes).

[2] Mishra, S., Tiwari, P., Yadav, R., & Patel, P. (2024). An Extensive Analysis of DiseasesAssociated with Diabetes. *2*(3), 174–187. [JOPIR](https://jopir.in/index.php/journals/article/view/181).

[3] Simaanjali. (n.d.). *Diabetes Simple Diagnosis* [Dataset]. [Kaggle](https://www.kaggle.com/datasets/simaanjali/diabetes-simple-diagnosis).