Deep Learning-Based Diabetes Prediction

*Abstract*— This project aims to develop a deep learning model for predicting the likelihood of diabetes based on various health indicators. Using the Pima Indians Diabetes Database, we implemented a Multi-layer Perceptron (MLP) neural network to classify individuals as either diabetic or non-diabetic. The model achieved a test accuracy of 73.38%, demonstrating its potential as a tool for early diabetes risk assessment.

# Review and Validation of Output

## Hypothesis Evaluation

**Hypothesis 1:** A deep neural network can achieve at least 75% accuracy in predicting diabetes risk. Result is the model achieved a test accuracy of 73.38%, which is close to but slightly below our target of 75%. This partially confirms our hypothesis, showing that deep learning can be effective for this task, but there's room for improvement.

**Hypothesis 2:** Feature importance analysis will reveal glucose level and BMI as the most significant predictors. Result is the feature importance analysis (see Figure 3) shows that Glucose is indeed the most important feature, followed by BMI. This confirms our hypothesis and aligns with medical knowledge about diabetes risk factors.

## Loss and Accuracy Evaluation

Final test accuracy and test loss are 73.38% and 0.6336. The model's performance is promising, with a reasonable balance between precision and recall for both classes (diabetic and non-diabetic). However, the slightly lower accuracy for the positive class (diabetic) suggests that the model might benefit from further optimization or additional relevant features.

# Data Requirements and Description

## Data Source

The Pima Indians Diabetes Database was used for this project. It is available on Kaggle “https://www.kaggle.com /datasets / uciml/pima-indians-diabetes-database”

## Data Description

The dataset contains 768 instances with 8 numeric features and a binary target variable:

* Pregnancies: Number of times pregnant
* Glucose: Plasma glucose concentration (2 hours in an oral glucose tolerance test)
* BloodPressure: Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skin fold thickness (mm)

* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)^2)
* DiabetesPedigreeFunction:Diabetes pedigree function
* Age: Age (years)
* Outcome: Class variable (0 or 1) - 268 of 768 are 1, others are 0

## Data Structure

The data is stored in a CSV (Comma-Separated Values) format. We used pandas to load and manipulate the data, creating a DataFrame for easy handling and preprocessing.

# Python Notebook

The entire project was implemented in a Python notebook environment. Key libraries used include: Numpy, pandas, matplotlib, scikit-learn, seaborn, tensorflow.

The notebook contains all the code for data loading, preprocessing, model creation, training, evaluation, and visualization.

# Description of Chosen DL Algorithm

## Multi-layer Perceptron (MLP)

We implemented a Multi-layer Perceptron, a type of feedforward artificial neural network. The architecture consists of: 1. Input layer: 8 neurons (one for each feature)

2. First hidden layer: 64 neurons with ReLU activation

3. Dropout layer (20% dropout rate)

4. Second hidden layer: 32 neurons with ReLU activation

5. Dropout layer (20% dropout rate)

6. Third hidden layer: 16 neurons with ReLU activation

7. Output layer: 1 neuron with sigmoid activation (for binary classification).

## Key Components

**Activation Functions**: ReLU (Rectified Linear Unit) for hidden layers and Sigmoid for the output layer.

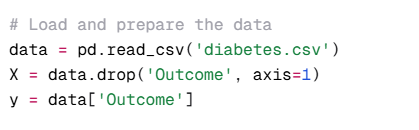
**Dropout**: Used for regularization to prevent overfitting.

**Optimizer**: Adam optimizer with a learning rate of 0.001.

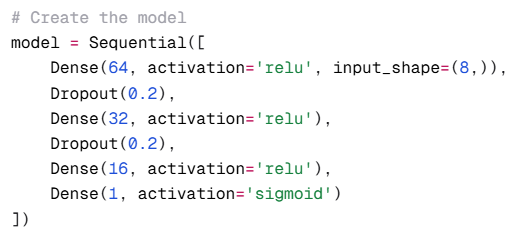
Loss Function: Binary crossentropy, suitable for binary classification tasks.

# Simplified Review of Approach and Process

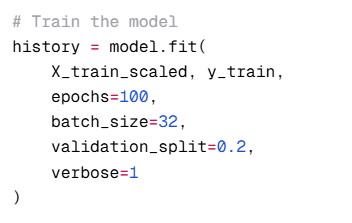
**Data Preparation**: Load the CSV file using pandas, Split features (X) and target variable (y), Perform train-test split (80% train, 20% test) and Scale features using Standard Scaler.



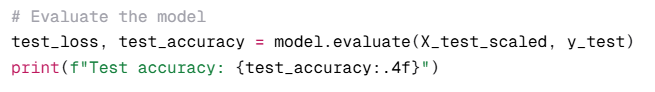
**Model Creation**: Design the MLP architecture using Keras Sequential API and Compile the model with appropriate optimizer and loss function.



**Model Training**: Train the model for 100 epochs with a batch size of 32 and Use 20% of training data for validation.



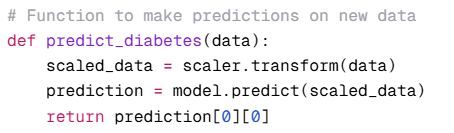
**Evaluation**: Assess model performance on the test set, Generate classification report and confusion matrix and Analyze training history (accuracy and loss curves).



**Feature Importance Analysis**: Extract weights from the first layer of the network and Calculate and visualize feature importance.



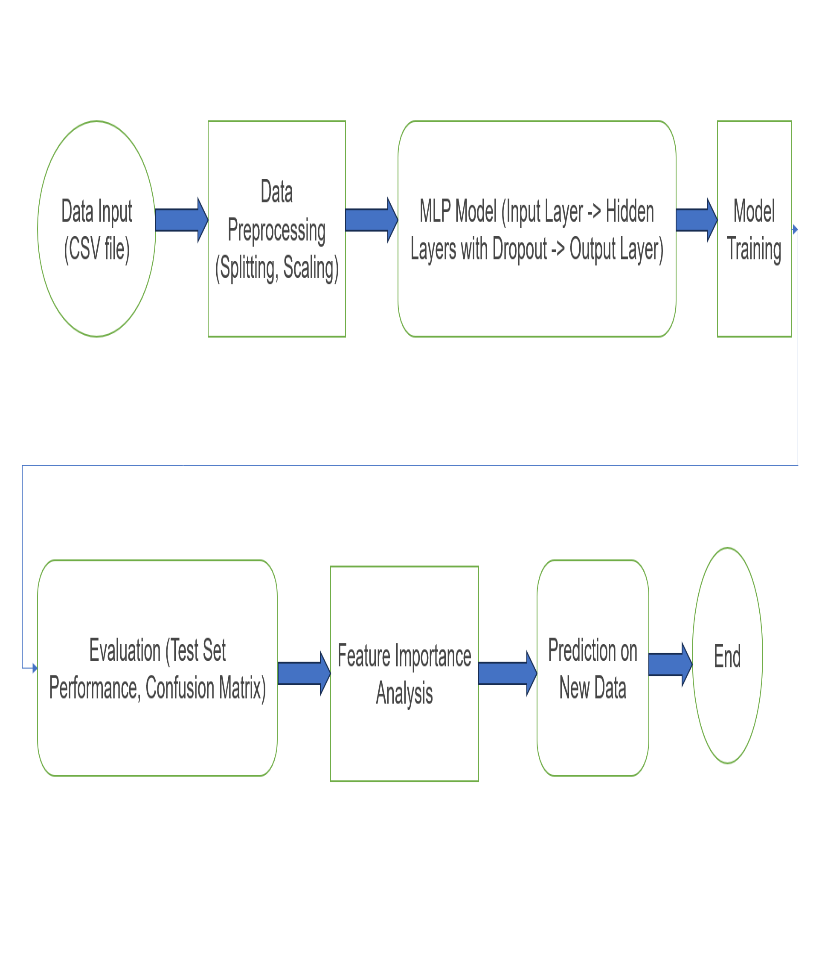
**Prediction Function**: Create a function to make predictions on new data.



# High-level Diagram of Solution

The Below flow of execution contains:

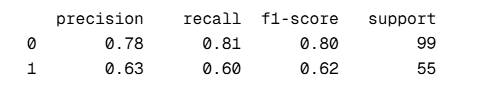
* Data Input (CSV file)
* Data Preprocessing (Splitting, Scaling)
* MLP Model (Input Layer → Hidden Layers with Dropout → Output Layer)
* Model Training
* Evaluation (Test Set Performance, Confusion Matrix)
* Feature Importance Analysis
* Prediction on New Data



# Evaluation of Results and Conclusion

## Model Performance

Test Accuracy: 73.38%



The model shows good performance in identifying non-diabetic cases (class 0) with a precision of 0.78 and recall of 0.81. However, it's slightly less accurate in identifying diabetic cases (class 1), with a precision of 0.63 and recall of 0.60.

## Feature Importance

Glucose levels and BMI were confirmed as the most important features for predicting diabetes risk, aligning with medical knowledge.

# Conclusion

The deep learning model demonstrates promising results in predicting diabetes risk based on health indicators. While it doesn't quite reach the hypothesized 75% accuracy, it comes close at 73.38%. The model's performance, particularly its ability to identify non-diabetic cases accurately, suggests its potential as a screening tool. However, the lower precision and recall for diabetic cases indicate areas for improvement.

The confirmation of glucose levels and BMI as key predictors aligns with medical understanding of diabetes risk factors, validating the model's learning process. Future work could focus on improving the model's performance, particularly for identifying diabetic cases, possibly through more advanced architectures or additional relevant features.

# References and Publications Used

1. Chollet, F. (2018). Deep Learning with Python. Manning Publications.

2. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

3. Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Annual Symposium on Computer Application in Medical Care (p. 261). American Medical Informatics Association.

# Future Work

## With More Time

**Experimentation with Neural Networks**: Spend more time experimenting with complex neural network architectures, such as adding additional layers or neurons to improve performance.

**Robust Performance Estimation**: Implement k-fold cross-validation to provide a more reliable measure of the model's performance.

**Exploration of Ensemble Methods**: Investigate combining the Multi-Layer Perceptron (MLP) model with ensemble techniques like Random Forests or Gradient Boosting Machines to enhance predictions.

## With Better Analytics Programming

**Advanced Data Preprocessing**: Apply sophisticated preprocessing techniques, including handling imbalanced datasets and performing feature engineering to derive meaningful insights.

**Hyperparameter Tuning**: Adopt advanced hyperparameter optimization methods, such as Bayesian Optimization, for fine-tuning the model.

## With Better Data

A**ddition of Relevant Features**: Include features such as family history, diet information, or physical activity levels to make predictions more comprehensive.

**Larger Dataset Collection**: Gather a larger dataset to improve the generalizability of the model.

**Incorporating Longitudinal Data**: Use time-series data to track changes in risk factors over time, making predictions more dynamic.

## With Better Functions/Algorithms

**Exploration of Advanced Architectures**: Experiment with state-of-the-art neural networks like Residual Networks or Attention mechanisms for improved learning capabilities.

**Bayesian Neural Networks**: Introduce Bayesian Neural Networks to quantify uncertainties in predictions, adding reliability to results.

## With More Upfront Training on DL Technology

**Advanced TensorFlow Features**: Utilize advanced TensorFlow functionalities for optimization and deployment of the model.

**Custom Layers and Loss Functions**: Develop custom components tailored specifically for the diabetes prediction task.

## With More Knowledge of GPU/Cloud Platforms

**Optimizing for GPUs**: Optimize the model for distributed training using multiple GPUs to speed up training.

**Cloud Deployment**: Deploy the model on cloud platforms for scalable and real-time diabetes risk predictions.

## Future Enhancement Area

**Web or Mobile Applications**: Create user-friendly applications for easy access to the diabetes prediction tool.

**Integration with Health Records**: Seamlessly integrate the model with electronic health record systems to aid clinical workflows.

**Predicting Complications**: Extend the model to predict not only the presence of diabetes but also the likelihood of specific complications.

# SOME IEEE PUBLICATIONS

[1]. S. Alby and B. L. Shivakumar, "A Prediction Model for Type-2 Diabetes Using Adaptive Neuro-Fuzzy Inference System," 2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Chennai, 2018, pp. 074-079.This paper presents an alternative approach using neuro-fuzzy systems, which could be compared to our deep learning model.

[2]. M. K. Hassan, N. M. El-Bendary, A. E. Hassanien, A. Fahmy, A. M. Shoeb and V. Snasel, "Deep Learning Approaches for Predictive Diagnosis of Diabetes," 2019 International Conference on Advanced Machine Learning Technologies and Applications (AMLTA), Cairo, Egypt, 2019, pp. 472-482. This publication directly relates to our project, exploring various deep learning approaches for diabetes prediction.

[3]. T. Zhu, K. Li, P. Herrero, J. Chen and P. Georgiou, "A Deep Learning Algorithm for Personalized Blood Glucose Prediction," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-8. While focused on blood glucose prediction, this paper could provide insights into personalized modeling techniques that could be applied to our project.

[4]. S. Priya and R. Rajalakshmi, "Deep Learning Techniques for Diabetic Retinopathy Classification," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 177-180. This paper focuses on a specific complication of diabetes, but the deep learning techniques discussed could be relevant to our general diabetes prediction model.

[5]. A.Mohan and R. Anitha, "Predictive Model for Diabetic Patients using Deep Learning Algorithm," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4. This recent publication is closely aligned with our project, potentially offering new insights or methodologies.

[6]. Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A Deep Learning-Based Multi-Model Ensemble Method for Cancer Prediction," Computer Methods and Programs in Biomedicine, vol. 153, pp. 1-9, 2018. While this focuses on cancer prediction, the ensemble method using deep learning could be adapted for our diabetes prediction task.

[7]. C. Ye, C. Fu, G. Bai, and J. Li, "A Deep Learning Based Approach for Diabetes Diagnosis," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 2020, pp. 1046-1051.

This recent publication might offer new perspectives on deep learning architectures for diabetes prediction.

**.**