Predicting Diabetes Using Artificial Neural Networks

**By Alireza Khandan**

# Implementation:

## Report Background

Diabetes mellitus is a significant global health challenge, necessitating predictive models to accurately identify individuals at risk. This project leverages artificial neural networks (ANNs) to predict diabetes within the Iraqi population using data from Medical City Hospital and Al-Kindy Teaching Hospital. The goal is to aid healthcare providers in early diagnosis and management, potentially reducing the burden of diabetes through timely interventions.

## Justification of Tools and Libraries Used

* **Google Colab**: Chosen for its integration with Google Drive, free GPU access, and compatibility with popular Python libraries essential for processing and analyzing large datasets efficiently.
* **TensorFlow and Keras**: These libraries are at the forefront of deep learning, providing tools and resources to build and deploy AI models efficiently. Keras, integrated within TensorFlow, offers a high-level API for rapid prototyping.
* **Pandas and NumPy**: Utilized for data manipulation and numerical computations. Pandas offers intuitive data structures for cleaning and analysis, while NumPy provides mathematical functions for operating on large arrays and matrices.
* **Scikit-learn**: Employed for robust scaling, transformation, and dataset splitting tools, crucial for preparing the dataset for the neural network.
* **Matplotlib and Seaborn**: Used for data visualization, aiding in understanding data distributions and model performance.

## Design / Data Preparation

The dataset includes medical and laboratory analysis data for 1,000 patients, classified into Diabetic, Non-Diabetic, and Predicted-Diabetic categories. Proper data preparation was critical to the model's performance.

## Data Cleaning and Preprocessing:

* **Initial Cleaning**: Addressed missing values.
* **Numerical Data**: Imputed missing values using column medians to reduce outlier effects.
* **Categorical Data**: Filled using the most frequent values to maintain the integrity of non-numerical information.
* **Standardization and Encoding**: Standardized numerical data and one-hot encoded categorical data to prepare for neural network processing, avoiding multicollinearity.
* **Feature Scaling**: Scaled numerical features using StandardScaler to ensure equal contributions to the analysis, enhancing model stability and performance.

## Implementation of the System/Model

The neural network, constructed with TensorFlow and Keras, includes:

* **Input Layer**: Accepts preprocessed data with inputs corresponding to health indicators.
* **Hidden Layers**: Multiple dense layers with ReLU activation to process data non-linearly, enhancing convergence.
* **Output Layer**: Uses a softmax function for multi-class classification, outputting probabilities for each class (Non-Diabetic, Predicted-Diabetic, Diabetic).
* **Optimization and Loss Function**: The Adam optimizer was chosen for its efficiency in handling sparse gradients and adaptive learning rates. The loss function used was categorical crossentropy, with accuracy as the performance metric.

## Results Related to Implementation

Training over 50 epochs showed significant accuracy improvement:

* **Training Accuracy**: Up to 98.2%
* **Validation Accuracy**: Around 91.2%
* **Test Accuracy**: 93.5% Validation loss stabilized, indicating the model did not overfit the training data significantly.

Example of Training Output:

A screenshot of a computer screen

Description automatically generated

# Reflection on the Implementation Process

Challenges included handling data variability and managing model complexity. Key lessons emphasized the importance of robust preprocessing and continuous performance evaluations to ensure the model's generalization capability.

# Recommendations for Future Enhancements or Improvements

* K-fold Cross-Validation: To ensure model robustness.
* Advanced Regularization Techniques: To further mitigate overfitting.
* Exploration of Ensemble Methods: Combining multiple models to improve prediction accuracy.
* Continuous Data Acquisition: Regular updates to the dataset to enhance model accuracy and relevance.

# Files in The Repository

* **Dataset of Diabetes.csv**: The dataset used to train the model, containing medical and laboratory analysis data.
* **Predicting Diabetes Using AI.ipynb**: The Jupyter notebook with code and explanations for the entire process, from data loading to model evaluation.
* **Predicting Diabetes Using AI.py**: The Python script containing the code used to create the project.
* **diabetes\_prediction\_model.keras**: The trained Keras model file saved in the '.keras' format, created by Block 6. This file allows you to load the trained neural network model without retraining it, enabling you to make predictions directly.
* **README.md**: This README file providing an overview of the project, usage instructions, and file descriptions.
* **Documentation (coming soon)**: Additional documentation explaining the project in more detail.

# Links:

Check it on my Google Colab: <https://colab.research.google.com/drive/1qW6D6Fp0OppmyAIib3gppQWsCRmWHmJc?usp=sharing>

Access to sourcecode and files on GitHub: <https://github.com/AlirezaKhandan/Predicting-Diabetes-Using-Artificial-Neural-Networks>