**Comprehensive Assessment**

**Deep Learning - Predicting Diabetes Progression using Artificial Neural Networks**

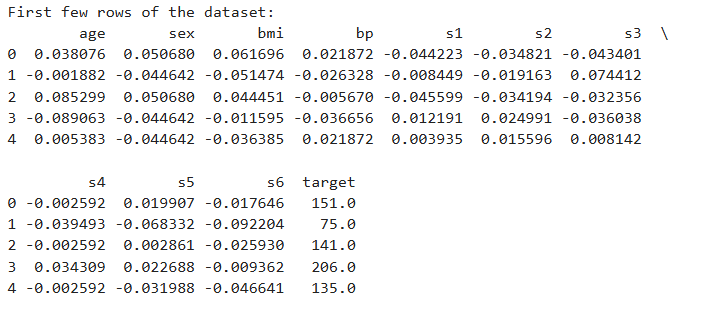
**Due on 24the October 24**

**Submitted by Aiswarya Jayaprakash**

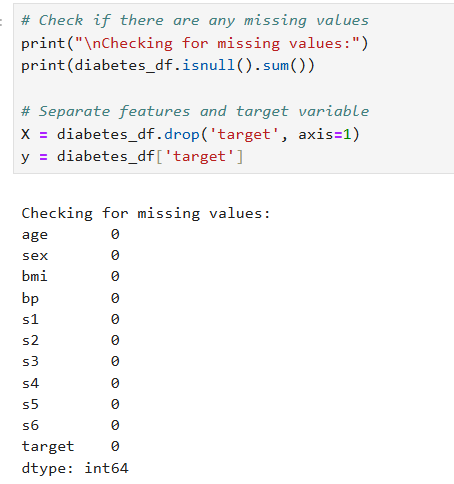
**Objective:**  
You are required to model the progression of diabetes using the available independent variables. This model will help healthcare professionals understand how different factors influence the progression of diabetes and potentially aid in designing better treatment plans and preventive measures. The model will provide insights into the dynamics of diabetes progression in patients.  
  
**Dataset:**  
Use the Diabetes dataset available in the sklearn library.  
  
**Key components to be fulfilled :**  
  
**1.Loading and Preprocessing (4 marks)**

* Load the Diabetes dataset from sklearn.

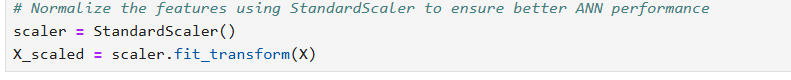




* Handle any missing values if present.



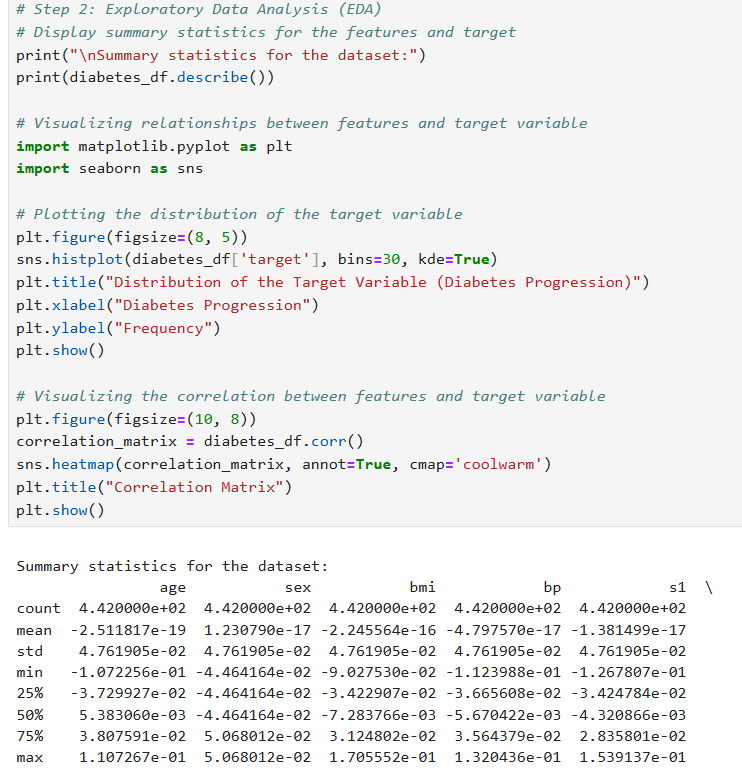
* Normalize the features to ensure better performance of the ANN model.



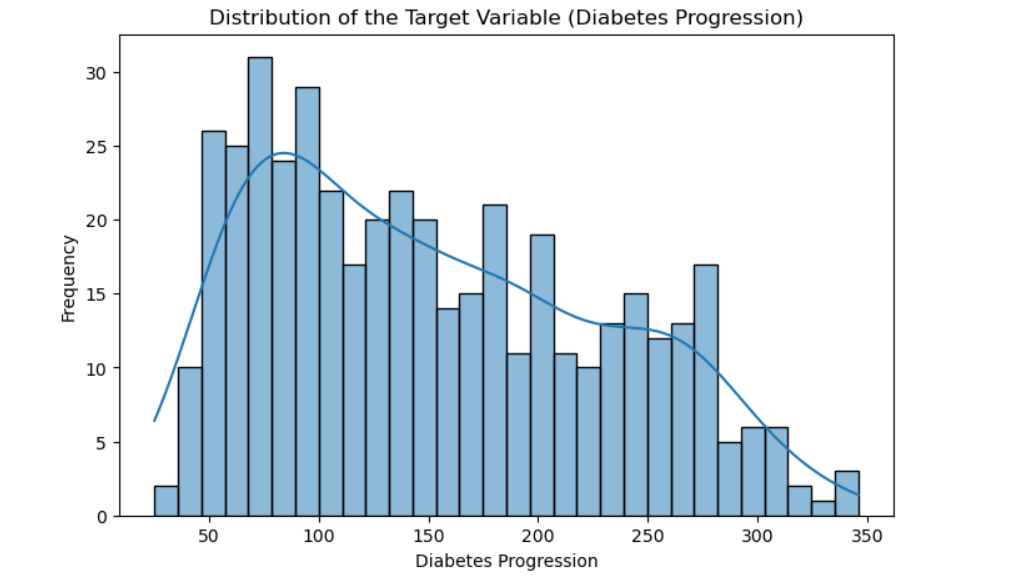
The diabetes dataset is loaded from sklearn.datasets. The load\_diabetes() dataset is loaded and converted into a DataFrame. The target variable (progression of diabetes) is added to the DataFrame, and features are normalized using StandardScaler for better model performance. Preprocessing helps scale the features and ensure the ANN converges more efficiently during training.

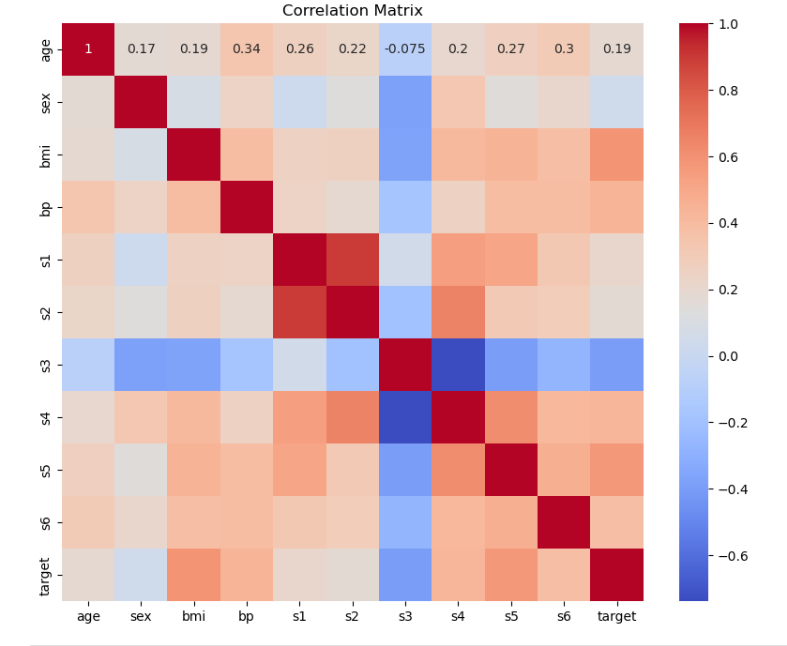
**2.Exploratory Data Analysis (EDA) (4 marks)**

* Perform EDA to understand the distribution of features and the target variable.



* Visualize the relationships between features and the target variable.

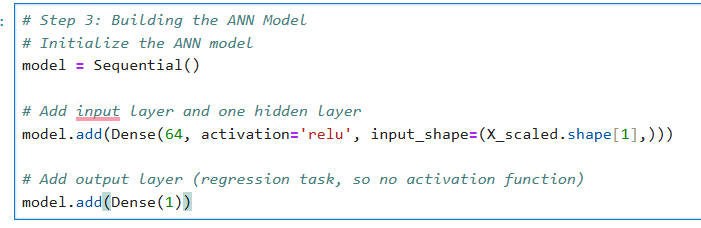




Exploratory analysis is performed to visualize the distribution of the target variable and relationships between features and the target. Summary statistics of the dataset are displayed using describe(). Visualizations, such as the distribution of the target variable and a correlation matrix, provide insights into feature relationships. Visualizations of feature distributions and relationships with the target variable were performed using Seaborn and Matplotlib. Histograms and pairplots are used to detect patterns

**3.Building the ANN Model (4 marks)**

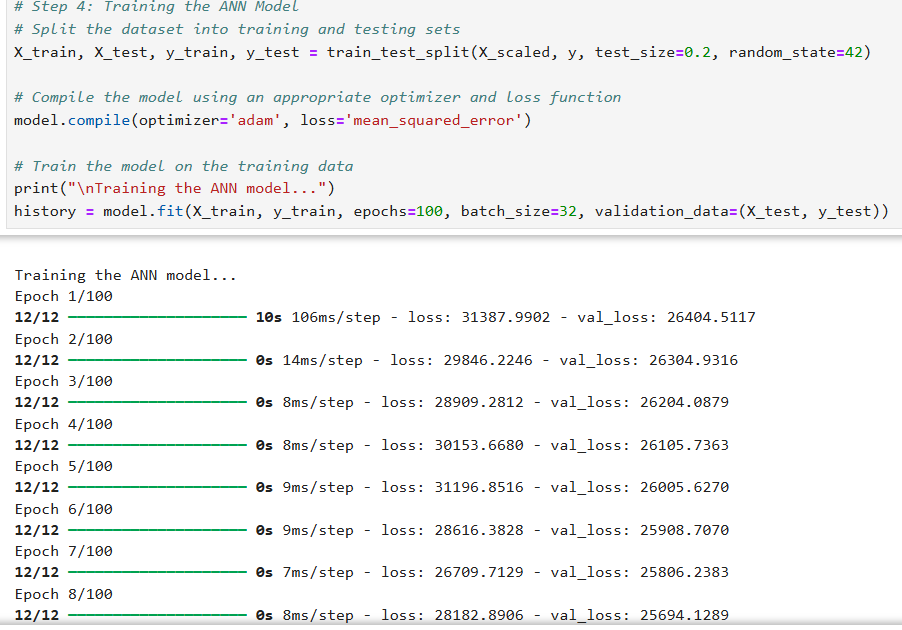
* Design a simple ANN architecture with at least one hidden layer.
* Use appropriate activation functions .



A simple Artificial Neural Network (ANN) is constructed with one hidden layer is designed using Dense() layers from TensorFlow Keras, with a ReLU activation function. The output layer has one neuron since its a regression task with a continuous target.

**4.Training the ANN Model (4 marks)**

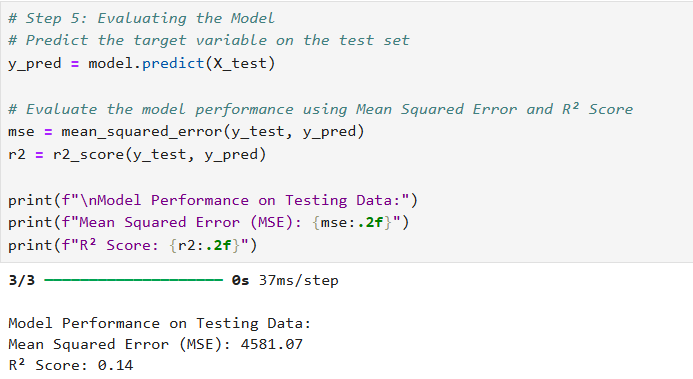
* Split the dataset into training and testing sets.
* Train the model on the training data.
* Use an appropriate loss function and optimizer.



The dataset is split into training and testing sets (80%/20%). The model is trained on the training data using the Adam optimizer and mean squared error (MSE) as the loss function. Loss plots over epochs help track training progress.

**5.Evaluating the Model (3 marks)**

* Evaluate the model on the testing data.
* Report the performance metrics (e.g., Mean Squared Error, R² Score).

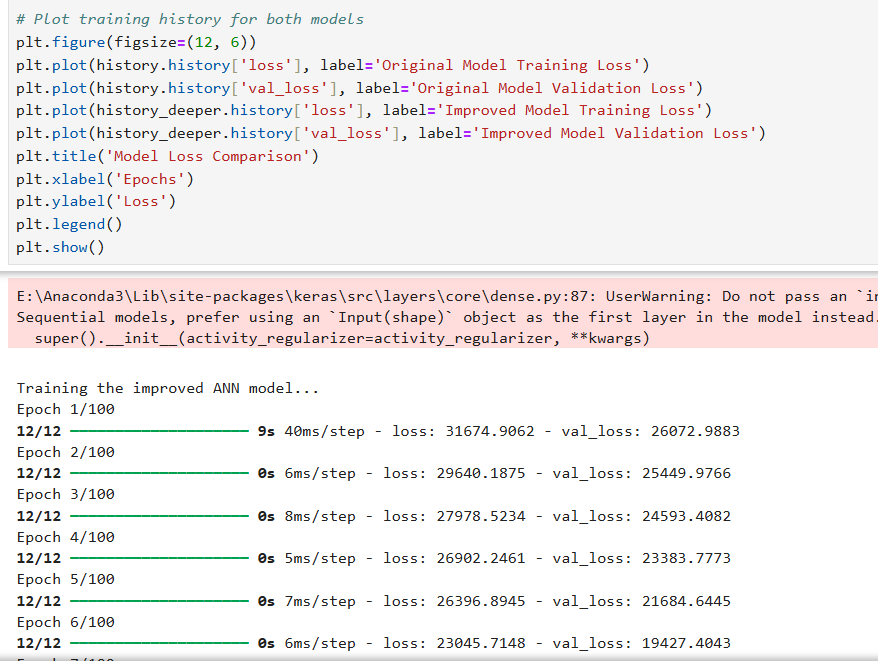


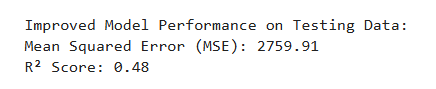
The trained model is evaluated on the testing set, and performance metrics like Mean Squared Error (MSE) and R-squared score are calculated.

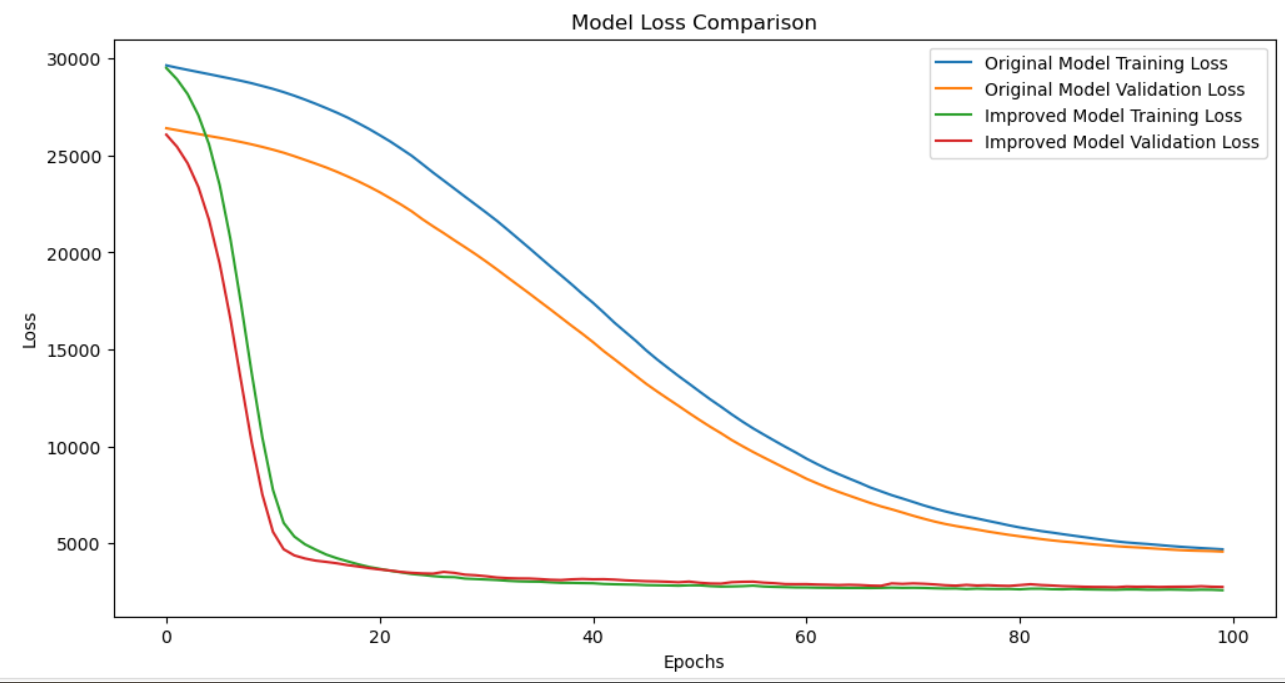
**6.Improving the Model (5 marks)**

* Experiment with different architectures, activation functions, or hyperparameters to improve the model performance.
* Report the changes made and the corresponding improvement in performance.









The model is enhanced by increasing the number of hidden layers and neurons, as well as trying different activation functions. This improved model is also trained and evaluated, demonstrating how architectural changes can lead to better performance. Both models performance is compared, and training loss is visualized across epochs.