Stroke Detection

By: Dalal Qallajo

Description: In this project, I developed a machine learning model to predict the likelihood of stroke occurrence based on various health-related features. The dataset used for this analysis, the Healthcare Dataset Stroke Data, was downloaded from Kaggle and includes critical information such as demographic details, health conditions, and lifestyle factors.

Key Features:

- Data Preprocessing:

Handled missing values in the BMI column.

Applied one-hot encoding to categorical variables.

- Model Development:

Implemented a neural network using TensorFlow and Keras.

Compiled the model with the Adam optimizer and binary cross-entropy loss.

- Model Training and Evaluation:

Split the dataset into training and testing sets.

Trained the model for 50 epochs and evaluated accuracy

- Performance Metrics:

Generated a confusion matrix.

Plotted the ROC curve and calculated AUC.

Calculated precision and recall scores.

- Feature Importance:

Analyzed feature importance using neural network weights.

Key Findings:

*Evaluation Results:

Loss: 0.15

Accuracy: 0.95

*Training Results:

Loss: 0.17 at epoch 50

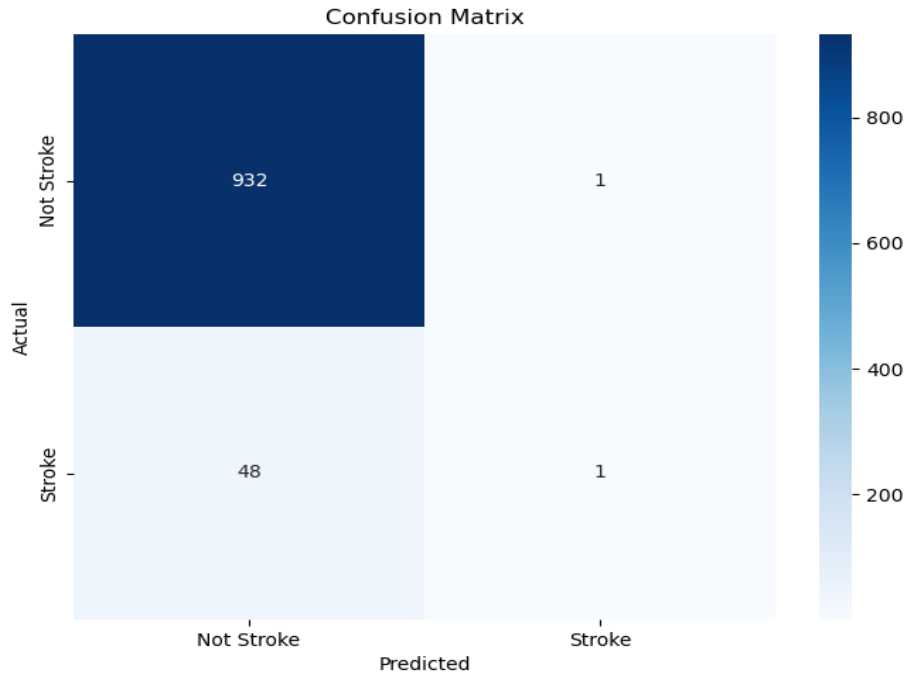
Accuracy: 0.96 at epoch 50

Visualization:

Confusion Matrix

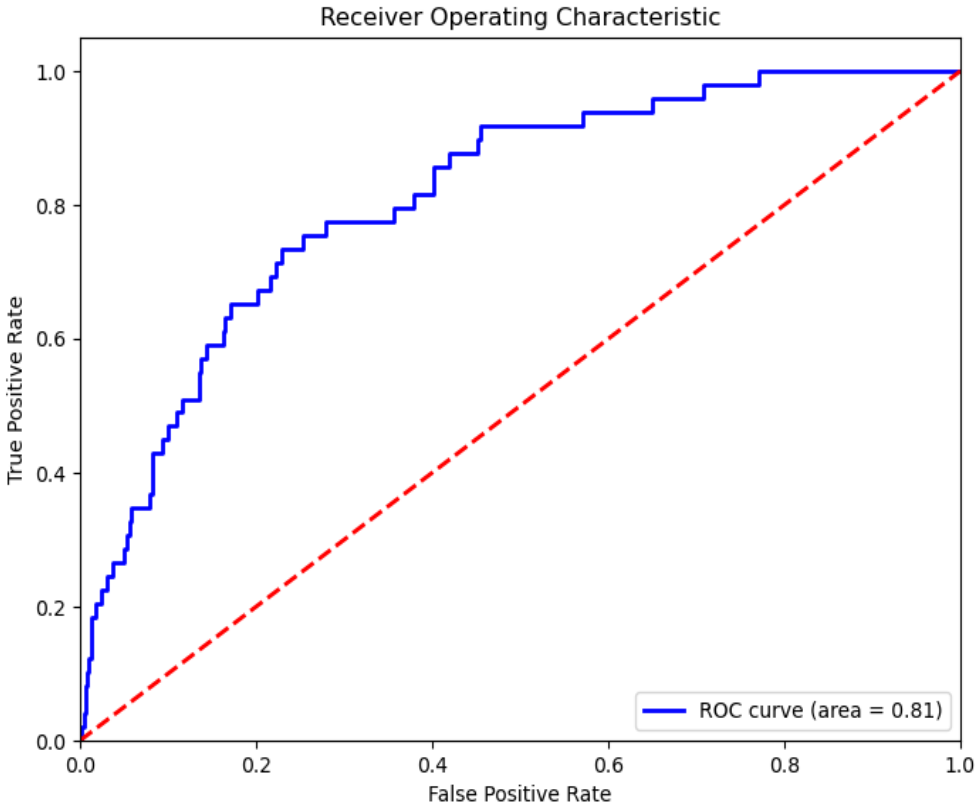
- The confusion matrix is a performance measurement tool for classification models, providing a comprehensive view of how well the model is performing. It summarizes the results of predictions made by the model against the actual outcomes.

- In the context of this project, the confusion matrix helps to visualize the performance of the stroke prediction model by displaying the following:
True Positives (TP)
True Negatives (TN)
False Positives (FP)
False Negatives (FN)



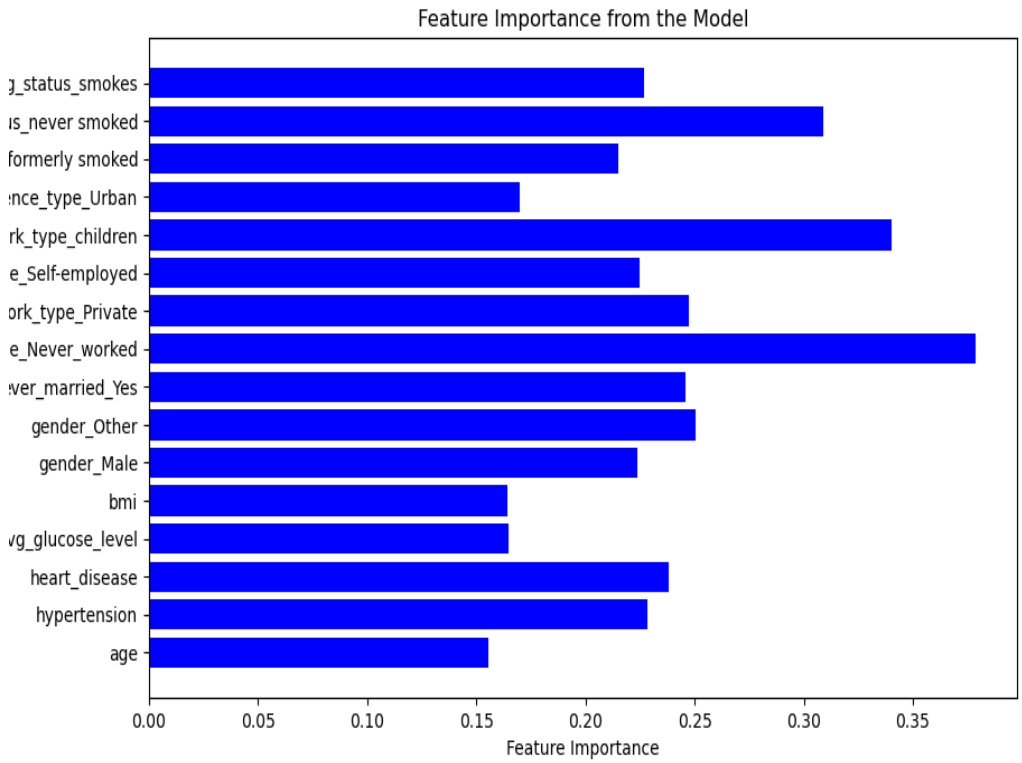
ROC curve

- ROC curve illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (specificity) at various threshold settings.
- The AUC quantifies the overall performance of the model. In this project, the AUC was calculated to assess how well the model distinguishes between stroke and non-stroke cases.
- By analyzing the ROC curve, we can better understand the model's performance and make informed decisions about its applicability in real-world scenarios.



Feature Importance

- This technique helps identify the features that contribute most to the model's decision-making process.
- In this project, feature importance was analyzed using neural network weights.



Code link on GitHub:

https://github.com/Dalalkaljo/Deep-Learning/blob/main/Stroke-Detection-Classification