**Neural Network Model Performance Analysis**

**1. Overview of the Analysis**

The purpose of this analysis is to evaluate the performance of three different neural network architectures in solving a binary classification problem. Each model was built using the TensorFlow Keras Sequential API and differs primarily in the choice of activation function used in the hidden layers: ReLU, ELU, and Swish. The goal is to determine which model performs best in terms of accuracy and loss, and to explore potential ways to improve future model performance.

**2. Results**

**A. Data Preprocessing**

* **Target Variable(s):**
  + A single binary classification label predicted using a sigmoid activation in the final layer.
* **Feature Variable(s):**
  + All columns in the dataset X are used as input features, with input dimension equal to len(X.columns).
* **Variables to Remove:**
  + Any non-numeric, identifier, or redundant columns that are neither part of X nor the target label should be removed prior to training (e.g., IDs, timestamps).

**B. Compiling, Training, and Evaluating the Model**

**Model 1 – ReLU Activation**

nn\_model\_1.add(Dense(units=5, activation="relu", input\_dim=len(X.columns)))

nn\_model\_1.add(Dense(units=3, activation="relu"))

nn\_model\_1.add(Dense(units=1, activation="sigmoid"))

* **Performance:**
  + Accuracy: **51.46%**
  + Loss: **0.6912**
* **Comment:**
  + Shows near-random performance, indicating limited learning capacity.

**Model 2 – ELU Activation**

nn\_model\_2.add(Dense(units=5, activation="elu", input\_dim=len(X.columns)))

nn\_model\_2.add(Dense(units=3, activation="elu"))

nn\_model\_2.add(Dense(units=1, activation="sigmoid"))

* **Performance:**
  + Accuracy: **53.72%**
  + Loss: **0.7112**
* **Comment:**
  + Slightly better accuracy but higher loss suggests possible instability or overfitting.

**Model 3 – Swish Activation**

nn\_model\_3.add(Dense(units=5, activation="swish", input\_dim=len(X.columns)))

nn\_model\_3.add(Dense(units=3, activation="swish"))

nn\_model\_3.add(Dense(units=1, activation="sigmoid"))

* **Performance:**
  + Accuracy: **49.84%**
  + Loss: **0.8382**
* **Comment:**
  + Lowest performance among all three models.

**Model Architecture Summary**

* All models used **3 layers**: 2 hidden layers and 1 output layer.
* Hidden layers had **5 and 3 neurons** respectively.
* Activation functions tested: ReLU, ELU, Swish.

**Model Performance Goals**

* None of the models achieved satisfactory performance (typically expected >70% accuracy).

**Potential Steps for Improvement**

* Increase model complexity (more layers/neurons).
* Add regularization techniques (dropout, L2).
* Normalize or standardize input features.
* Handle class imbalance (e.g., SMOTE or class weights).
* Tune optimizer settings and learning rates.

**3. Summary**

The overall results of the neural network models were unsatisfactory, with all three models achieving accuracy close to random guessing (~50%). Among them, **Model 2 using ELU activation** performed the best with 53.72% accuracy, though still far from a usable classifier.

**Recommendation:**

It is recommended to explore alternative models, particularly:

* **Tree-based Models (e.g., Random Forest, XGBoost):**
  + These models often perform better on structured/tabular data.
  + They are less sensitive to feature scaling and often capture complex feature interactions.

Additionally, enhancements to data preprocessing, feature engineering, and tuning hyperparameters will likely contribute more to improved performance than simple changes in neural network architecture.