# **Assignment 1: Implementing Feedforward Neural Networks in Python using Keras and TensorFlow for Bank Customer Churning**

## **Problem Statement**

The problem is to build a **Feedforward Neural Network (FNN)** to predict **customer churn** (whether a customer will leave the bank or not) using the **Bank Customer Churn dataset**. Customer churn is a key business challenge for banks and financial institutions, as retaining existing customers is more cost-effective than acquiring new ones.

## **Objective**

* To preprocess and prepare the dataset for deep learning.
* To implement a feedforward neural network using **Keras and TensorFlow**.
* To train the model to classify whether a customer will churn (Exited = 1) or stay (Exited = 0).
* To evaluate model performance on unseen data.

## **Requirements**

* **Operating System**: Windows / Linux / macOS (Colab/Anaconda supported)
* **IDE / Platform**: Google Colab / Jupyter Notebook

### **Libraries and Packages Used**

* **TensorFlow / Keras** → Building and training the neural network
* **Pandas** → Data handling and preprocessing
* **NumPy** → Numerical computations
* **Matplotlib / Seaborn** → Visualization
* **Scikit-learn** → Preprocessing, encoding, splitting dataset, evaluation metrics

## **Theory**

### **Definition**

A **Feedforward Neural Network (FNN)** is the simplest form of artificial neural network where data flows in one direction — from input layer → hidden layers → output layer. There are no loops or feedback connections.

### **Structure**

1. **Input Layer** – Receives customer data (features such as age, credit score, balance, etc.).
2. **Hidden Layers** – Intermediate processing units that apply weights, biases, and activation functions to learn complex patterns.
3. **Output Layer** – Predicts the probability of churn (binary classification: 0 or 1).

## **Methodology**

1. **Data Preprocessing**
   * Load dataset and handle categorical variables (e.g., Gender, Geography) using encoding.
   * Normalize/scale numerical features.
   * Split dataset into training and test sets.
2. **Model Construction**
   * Define a feedforward neural network using Keras Sequential API.
   * Add input, hidden, and output layers with appropriate activation functions (ReLU for hidden layers, Sigmoid for output).
3. **Model Training**
   * Compile the model with an optimizer (Adam) and loss function (binary\_crossentropy).
   * Train on training data using mini-batch gradient descent.
4. **Evaluation**
   * Test model accuracy on unseen data.
   * Use confusion matrix, classification report, and accuracy score to evaluate performance.

## **Advantages**

* Captures **non-linear relationships** in customer data.
* Easy to implement using **Keras/TensorFlow**.
* Works well for binary classification problems.
* Can be extended to more complex deep learning models.

## **Limitations**

* Requires **large dataset** for better generalization.
* May **overfit** if not regularized properly.
* Interpretability is low compared to simpler models like decision trees.
* Sensitive to feature scaling and data preprocessing.

## **Working / Algorithm**

1. Initialize weights and biases randomly.
2. Perform **forward propagation**: inputs pass through layers with weighted sums and activation functions.
3. Calculate prediction error (loss) using binary cross-entropy.
4. Apply **backpropagation** to compute gradients of weights.
5. Update weights using **gradient descent optimizer**.
6. Repeat for multiple epochs until model converges.

## **Conclusion**

This assignment demonstrated the implementation of a **Feedforward Neural Network (FNN)** for predicting customer churn using Keras and TensorFlow. The model successfully learned from historical customer data and predicted churn probability on unseen data. While effective, further improvements can be made by experimenting with deeper architectures, dropout layers, or more advanced models like LSTMs or ensemble learning.