Predicting Bank Loan Approval using Artificial Neural Networks (ANN)

1. Executive Summary:

This project employs Artificial Neural Networks (ANN) to predict the likelihood of bank loan approval based on customer attributes. Utilizing Python libraries such as pandas, numpy, seaborn, matplotlib, TensorFlow/Keras, and scikit-learn, the analysis delves into the Universal Bank dataset. The journey unfolds through data exploration, preprocessing, model development, and evaluation.

2. Introduction:

This project aims to develop an Artificial Neural Network (ANN) model for predicting bank loan approval. Leveraging Python libraries such as pandas, numpy, seaborn, matplotlib, and TensorFlow/Keras for neural network development, the analysis centers on the 'UniversalBank.csv' dataset. This dataset encompasses customer attributes like age, income, credit card usage, personal loan status.

3. Data Source:

The dataset used for this project is sourced from *Kaggle* and is provided by *Srihari Promod.* It is designed for predictive modeling tasks related to bank loan approval. The dataset comprises customer attributes and loan approval indicators, providing a valuable resource for developing machine learning models, particularly classification algorithms.

Dataset Information:

- **ID:** Unique identifier for each customer.
- **Age:** Age of the customer.
- **Experience:** Number of years of professional experience.
- **Income:** Annual income of the customer.
- **ZIP Code:** ZIP code of the customer's residence.
- **Family:** Number of family members.
- **CCAvg:** Average credit card spending per month.
- **Education:** Education level of the customer (1: Undergraduate, 2: Graduate, 3: Advanced/Professional).
- **Mortgage:** Value of house mortgage, if any.
- **Personal Loan:** Binary indicator of whether the customer accepted a personal loan (1) or not (0).
- **Securities Account:** Binary indicator of whether the customer has a securities account (1) or not (0).
- **CD Account:** Binary indicator of whether the customer has a certificate of deposit (CD) account (1) or not (0).
- **Online:** Binary indicator of whether the customer uses online banking services (1) or not (0).
- **CreditCard:** Binary indicator of whether the customer owns a credit card issued by the bank (1) or not (0).

Dataset Link: *Bank Dataset on Kaggle*

3. Data Exploration and Preprocessing:

- **Dataset Overview:** The dataset is loaded and explored using pandas functions like pd.read_csv(), info(), and describe().
- **Exploratory Data Analysis (EDA):**
  - Descriptive statistics and visualizations with seaborn and matplotlib offer insights into feature distributions.
  - Key features, including age, income, education, credit card ownership, and personal loan status, are visualized to understand the dataset's characteristics.

4. Data Visualization:

- Matplotlib and Seaborn are employed for visually interpreting different aspects of the dataset. This includes count plots for categorical variables, histograms for numerical variables, and a heatmap for the correlation matrix.

5. Data Preprocessing:

- Data preprocessing involves handling missing values, scaling numerical features using StandardScaler from sklearn, and creating two dataframes for approved and unapproved personal loans.
- One-hot encoding is used for the target variable ('Personal Loan'), and the dataset is split into training and testing sets using sklearn's train_test_split().

6. Neural Network Model:

- A sequential Keras model is constructed, incorporating input, hidden, and output layers.
- Dropout layers are introduced to mitigate overfitting.
- Model compilation utilizes categorical crossentropy loss, the Adam optimizer, and custom metrics (including F1 score).
- Scikit-learn's StandardScaler is utilized for feature scaling.

7. Model Training:

- The ANN model is trained using the training dataset.
- The training process is monitored and visualized through the loss across epochs using Matplotlib.

8. Model Evaluation:

- Predictions on the test set are made, and sklearn's metrics module calculates various metrics, including F1 score, precision, recall, and accuracy.
- A confusion matrix, visualized with seaborn, provides an understanding of the model's performance in class differentiation.

9. Conclusion:

- The project offers a comprehensive approach to bank loan prediction using ANN, incorporating data exploration, preprocessing, model building, and evaluation.
- Visualizations and analyses facilitate a deeper understanding of the dataset and loan approval factors.

10. Recommendations and Future Work:

- Feature engineering and hyperparameter tuning, facilitated by sklearn, could enhance model performance.
- Further segmentation analysis could offer insights for targeted marketing strategies.

11. Acknowledgments and Credits:

- This report is submitted by **Rounik Mondal**, a B.Tech Computer Science and Engineering Student at Lovely Professional University, under the guidance and support of **Ayan Kumar Ghosh**. This Project was completed as a part of the coursework for the **"**Artificial Intelligence**"** course offered by InternsElite.
- For a detailed walkthrough and implementation of the Bank Loan Prediction, please refer to the associated Google Colab notebook: https://colab.research.google.com/drive/1BAst31bILBskDsrERiZZnlutg7lITGHD

11. References:

- https://pandas.pydata.org/docs/
- https://numpy.org/doc/stable/
- https://matplotlib.org/stable/index.html/
- https://www.tensorflow.org/api_docs/python/tf/all_symbols
- https://www.tensorflow.org/guide/keras
- https://scikit-learn.org/0.21/documentation.html
- https://www.kaggle.com/datasets/sriharipramod/bank-loan-classification/
- https://seaborn.pydata.org/api.html
- https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam
- https://www.tensorflow.org/api_docs/python/tf/keras/metrics/Accuracy
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/Activation
- https://www.tensorflow.org/api_docs/python/tf/keras/utils/to_categorical
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- https://www.tensorflow.org/api_docs/python/tf/keras/backend