PROJECT REPORT

DEEP LEARNING TECHNOLOGIES

Home Credit - Loan Default Risk Assessment Using Deep Learning

TEAM-7

Divya Neelamegam, Harika Boyina, Poojitha Venkat Ram, Shruti Badrinarayanan

1. Introduction

In the realm of consumer finance, predicting loan defaults with high precision is critical for lenders to manage risks effectively. Financial institutions increasingly rely on deep learning techniques due to their ability to handle complex, non-linear relationships inherent in financial data. Research highlights the importance of these methods: for example, a study [1] found that loan default rates can vary significantly, noting a reduction from 9.5% to 2% in certain regions due to improved screening and credit management practices.

The early detection of potential defaults is vital not only for minimizing losses but also for sustaining the health of the financial ecosystem. Surveys by financial regulators and academic studies underscore the potential of advanced predictive analytics in preempting financial delinquencies. Deep learning models leverage vast amounts of data and have been shown to significantly enhance the predictive accuracy of financial risk assessments, evidenced by performance improvements in practical applications [2].

Moreover, the financial landscape is continuously evolving with changes in economic conditions and borrower behaviour. It is crucial for predictive models to not only be accurate but also robust over time. Stability in these models ensures that they remain effective without requiring frequent recalibrations, which can be resource-intensive. This balance between accuracy and stability is key to the deployment of reliable financial risk assessments.

The application of neural networks in loan default prediction provides foundational insights into borrower behaviours and risk factors. Techniques like regularization and dropout are integrated into neural networks to prevent overfitting and enhance model generalization. These adjustments are crucial for maintaining the model's performance across diverse datasets and over time.

Furthermore, sophisticated deep learning approaches such as Bidirectional Long Short-Term Memory (Bi-LSTM) and DNN with residual block are particularly effective in capturing

temporal dynamics and sequences in data, which are indicative of a borrower's financial behaviour. These models offer deeper insights and higher predictive accuracy, thereby enabling lenders to make more informed and timely decisions.

By integrating advanced deep learning techniques into financial risk assessment, lenders and banks can enhance their ability to predict loan defaults. This not only improves financial outcomes but also supports greater financial inclusion by enabling more accurate and fair assessments of borrower risk.

2. Literature Survey / Related Work

A study [3] conducted by Muhamad Abdul Aziz Muhamad Saleh Jumaa and Mohammed Saqib in 2023 combines deep learning with traditional data mining techniques to significantly enhance the precision of credit risk assessments. This innovative approach has streamlined the review process for loan applications and drastically improved the default prediction rate. By utilizing a neural network framework built on TensorFlow, they successfully developed a model capable of distinguishing likely defaulters with an accuracy of 95.2%. This breakthrough demonstrates the profound impact that integrating advanced machine learning models can have on the financial sector’s ability to manage credit risk effectively.

In another groundbreaking work, Shasha Liu and her team applied a Bayesian deep learning framework [4] to tackle the complexities of loan default prediction. Their 2023 study addresses the perennial challenge of incomplete and noisy financial data by employing Bayesian methods, which are adept at managing uncertainty. By analyzing the Kaggle Lending Club loan dataset, their model not only adapted to the inherent data inconsistencies but also surpassed traditional predictive models with an accuracy rate exceeding 96%. This methodology promises a more reliable and nuanced approach to predicting financial risk.

A 2023 paper introduced the use of Multi-view Graph Convolutional Networks (GCNs) to enhance loan default risk prediction [5]. This novel approach by leveraging a multi-view framework allows the model to utilize auxiliary information from connected application records, particularly useful in handling datasets with many missing entries. The enhanced model outperforms both conventional statistical models and standard deep learning approaches, particularly in dealing with imbalanced datasets which is a common issue in financial data. This innovation marks a significant step forward in the application of graph-based deep learning techniques for financial risk assessment.

The research by Ebenezer Owusu and his colleagues in 2023 focuses on overcoming the challenge of imbalanced datasets in loan default prediction using deep learning [6]. They utilized the ADASYN algorithm to effectively balance the dataset, which is crucial for training robust predictive models. The deep neural network employed in their study was optimized to achieve a high prediction accuracy of 94.1%, demonstrating the effectiveness of synthetic data in enhancing model training and predictive performance. This work highlights the potential of tailored deep-learning solutions in addressing specific challenges in financial modelling. Read more about their findings.

Yanzhen Qu and Ihsan Said's 2023 research explores the application of both Convolutional and Recurrent Neural Networks to predict credit card defaults, highlighting the adaptability of deep learning models to complex data structures [7]. Their comparative analysis provides valuable insights into the strengths and limitations of each model type in processing and learning from financial data. By establishing a rigorous methodology for evaluating and comparing model performance, their work lays a solid foundation for future advancements in the field, ensuring that financial institutions can better harness the power of AI to predict and mitigate risks.

Adaleta Gicić and Dženana Đonko proposed a novel model in 2023 that combines deep learning methods with the SMOTE technique to tackle the challenge of imbalanced datasets in credit risk prediction [8]. By employing Stacked LSTM and Stacked BiLSTM architectures, their approach creatively adapts time series methodologies to the classification problems typical of credit scoring. This study underscores the importance of innovatively using deep learning to enhance predictive accuracy and adaptability in financial applications.

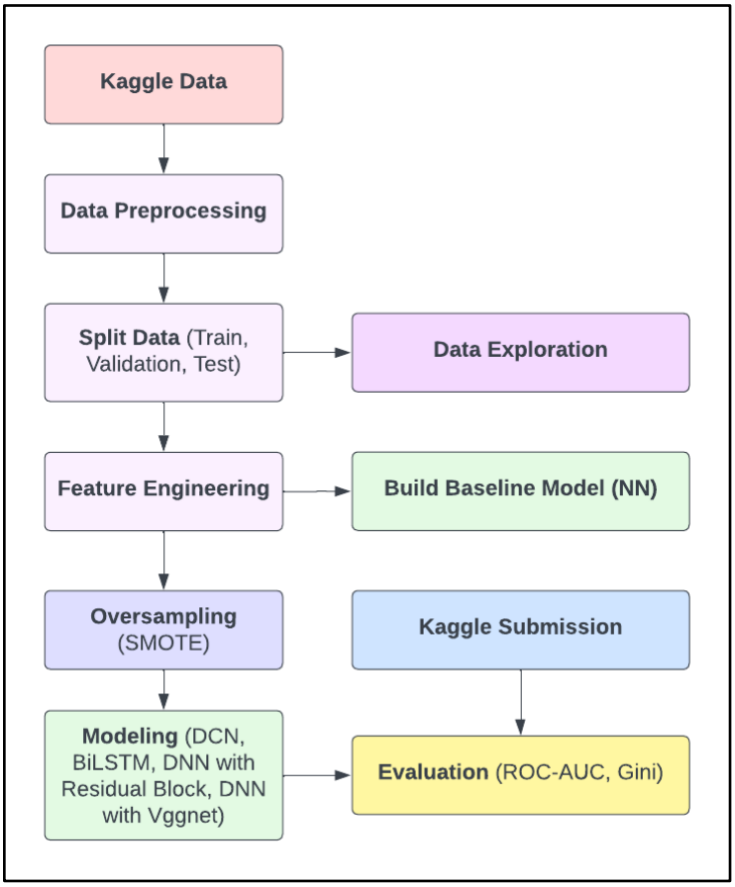
Shivaram Hegde's work on Uncertainty Quantification (UQ) in credit risk management using a Deep Evidence Regression approach represents a significant advancement in handling the uncertainties inherent in financial risk predictions. This 2023 study applies advanced deep learning techniques to model loss-given defaults, emphasizing the importance of incorporating uncertainty into predictive models [9]. This approach enhances the robustness and reliability of predictions, which is crucial for making informed risk management decisions in the financial sector.

3. Project Architecture

The project architecture for analyzing loan default risk using deep learning is designed to efficiently handle large-scale financial datasets and accurately predict borrower behavior. This strong framework illustrated in Figure 1 uses a variety of deep learning models and advanced data processing techniques to improve predictive performance and reliability in financial assessments.

Figure 1

System Architecture



The data pipeline is an essential component of our architecture. It begins with data collection from Kaggle, which includes demographics, loan activities, financial habits, and credit history. This data is then standardized and cleaned in preparation for analysis, with types unified, missing values imputed, and categorical variables encoded during the data preprocessing stage. Then splitting the data into train, test and validation sets performed exploratory data analysis, following that, feature engineering is performed, in which data from related tables is aggregated and new features are created based on domain knowledge to capture critical aspects of credit risk. Built a baseline neural network model. Statistical methods and dimensionality reduction are used in feature selection to refine the feature set, thereby improving model efficiency and performance.

Used the Oversampling technique (SMOTE) to oversample the minority target class(class 1) in the dataset.

During model development and evaluation, we first select various models to determine their predictive power and suitability for the task. Simple baseline neural networks, Deep Classification Networks (DCN), DNN with Residual block, BiLSTM and VGGNet-inspired tabular data models are all included. These models are trained over multiple epochs using techniques such as regularization, dropout, and batch normalization to prevent overfitting and ensure robust learning. The models are then evaluated using metrics like Area Under the Receiver Operating Characteristic Curve (AUROC), and Gini Stability to determine their effectiveness in distinguishing between default and non-default cases.

In the final phase, models are thoroughly compared and selected based on performance metrics. All the built models are submitted to the Kaggle competition. The best model is chosen based on its AUROC score, stability over time, and interpretability, ensuring that the chosen model not only performs well but also remains stable and understandable.

This architecture facilitates iterative testing and model refinement, allowing for systematic improvements in prediction accuracy as well as the ability to handle the complexities and scale of financial data effectively.

4. Data Exploration and Preprocessing

In Figure 2, according to the pie chart, class 0 accounts for 96.9% of the data, while class 1 accounts for 3.1%. The bar chart shows this distribution, with class 0 surpassing 1.5 million counts compared to a far smaller figure for class 1. In Figure 3, The graph depicts loan application counts per month for the Home Credit Risk competition, revealing a large fluctuation in monthly loan applications. The x-axis depicts the months, with blue bars representing loans with no payment issues (target '0') and orange bars representing loans with payment issues (target '1'). The loans with payment issues are less in all months. Seasonal patterns are evident with application peaks at the beginning and end of the year and declines in the middle months.

Figure 2

Total distribution of target variables in the dataset

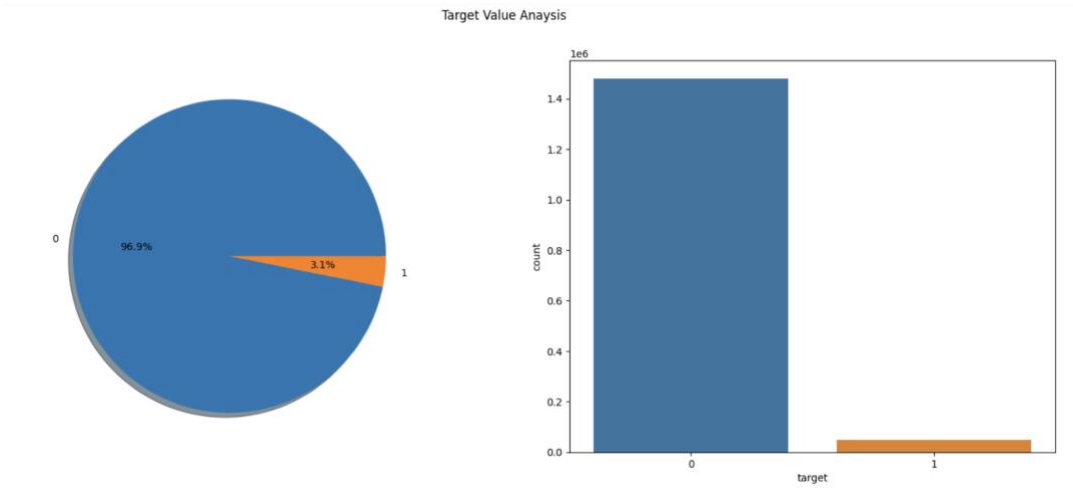


Figure 3

Monthly distribution of target variables

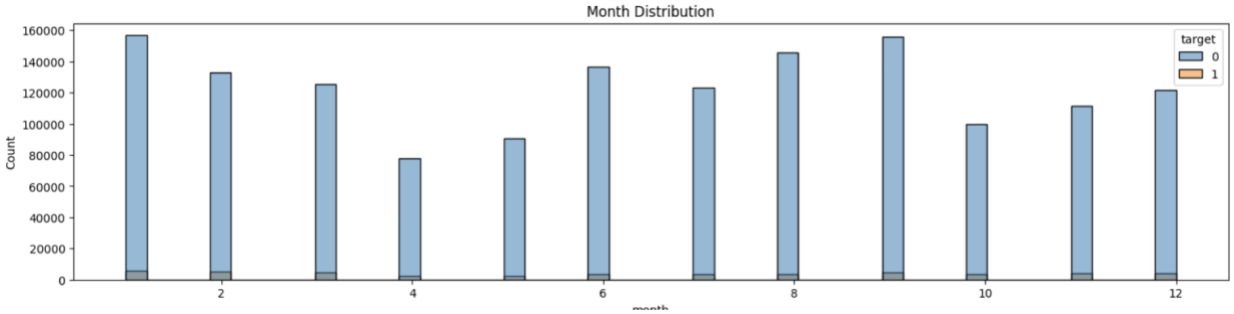


Figure 4 shows the frequency distribution of annuity amounts for a financial dataset. The graph displays a significant peak at lower values followed by a rapid decrease as the annuity amount grows, demonstrating that the majority of annuity payments are small, with few reaching greater sums. The distribution is highly right-skewed, indicating a concentration of values around the bottom of the scale. In Figure 5, the box plot for the feature "annuity_780A" depicts the distribution of annuity values in the data set. The tight box indicates that the vast majority of annuity values are concentrated in smaller quantities. The median value is low, and the interquartile range is slightly higher. The plot also reveals a large number of outliers, up to over 80,000, indicating that while the majority of annuities are low, there are a few unusually high amounts.

Figure 4

Distribution of feature: annuity_780A

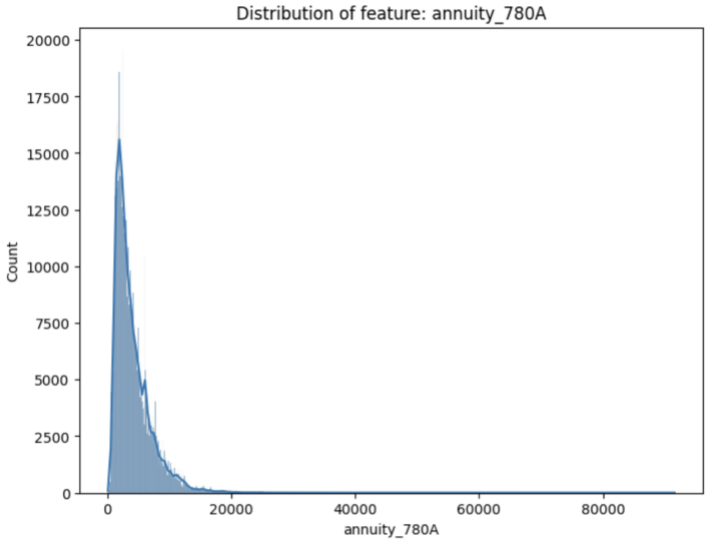
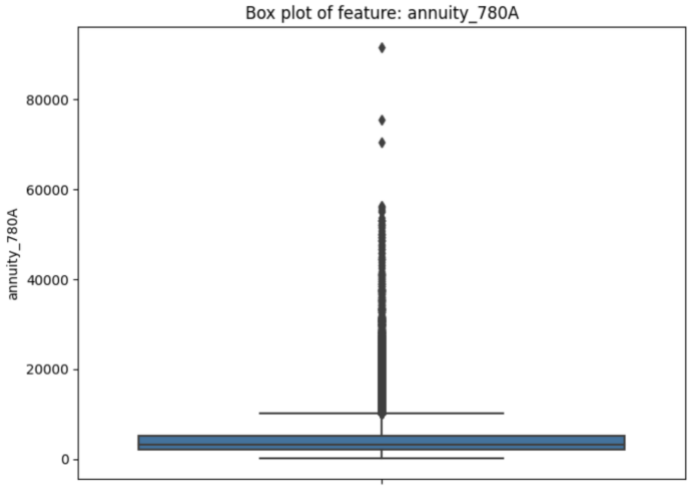


Figure 5

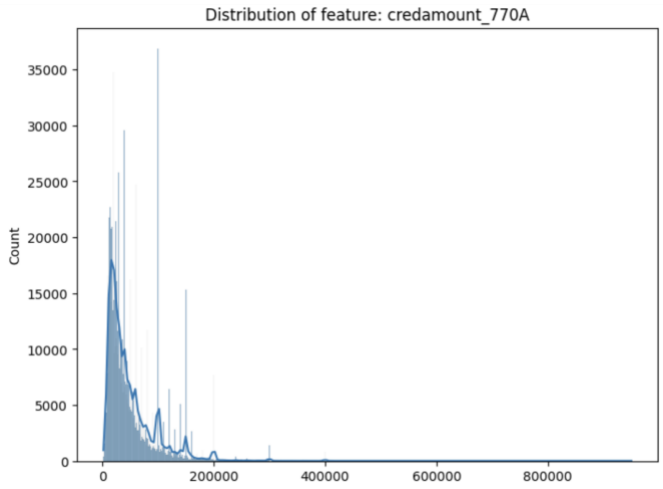
Box plot of feature: annuity_780A



In Figure 6, the histogram "Distribution of feature: credamount_770A" shows how credit amounts are distributed in the dataset. It demonstrates a strong concentration of lower credit amounts, with a peak near zero and a quick fall as the amount grows. The distribution is strongly right-skewed, with multiple minor peaks indicating potential credit amount intervals. There are also some extreme numbers, up to 800,000, that appear as outliers in this distribution.

Figure 6

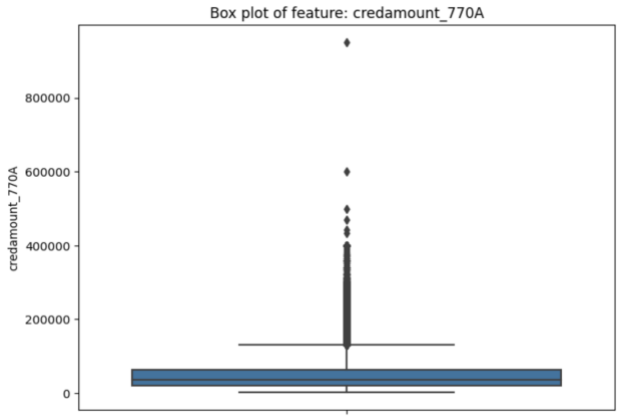
Distribution of feature: credamount_770A



The box plot in Figure 7, for the feature "credamount_770A" depicts the distribution of credit amounts in the dataset. The compact box showing the interquartile range depicts how the majority of credit amounts are focused at lower levels. The median is relatively low in the range of data. The plot also shows a substantial number of outliers, which extend well over the upper quartile, with some values exceeding 800,000, indicating extraordinarily high credit amounts on occasion.

Figure 7

Boxplot of feature: credamount_770A



The histogram in Figure 8, titled "Distribution of featurdisbursedcredamount_1113A" depicts the frequency distribution of disbursed credit amounts in a financial dataset. The distribution is highly right-skewed, with the majority of the data concentrated on smaller credit amounts and a dramatic peak near zero. A few higher values produce many smaller peaks further along the x-axis, although they are far

less common. The graph shows that, while the majority of disbursed credits are tiny, there are a few larger amounts, totalling 800,000.

Figure 8

Distribution of feature: disbursedcredamount_1113A

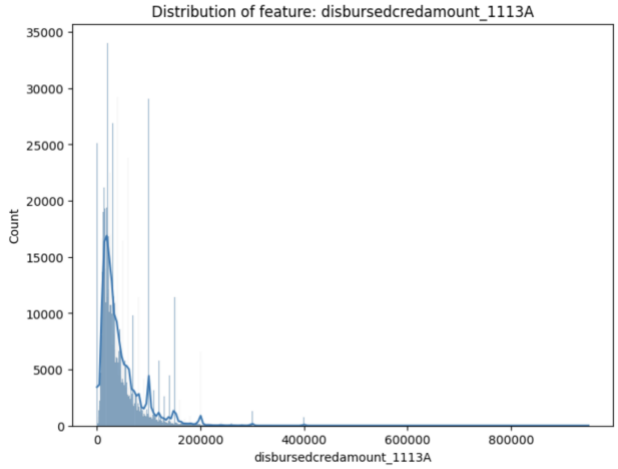
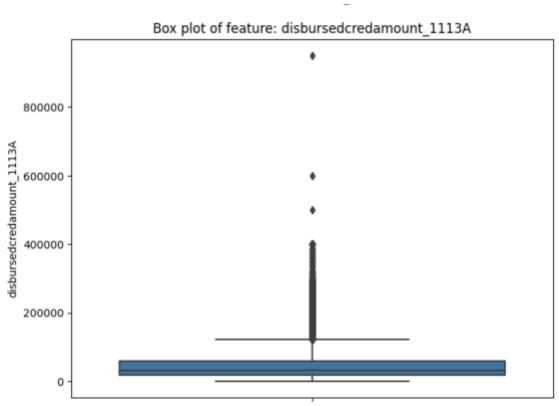


Figure 9 depicts the boxplot for the feature "disbursedcredamount_1113A" depicts how disbursed credit amounts are distributed. The majority of the credit amounts are concentrated at lower levels, as evidenced by the box's compactness, with the median and quartiles toward the lower end of the range. There are a lot of outliers, as indicated by the points above the main distribution that reach 800,000. This distribution emphasizes the majority of lower dispersed amounts, with a few relatively higher values.

Figure 9

Boxplot of feature: disbursedcredamount_1113A

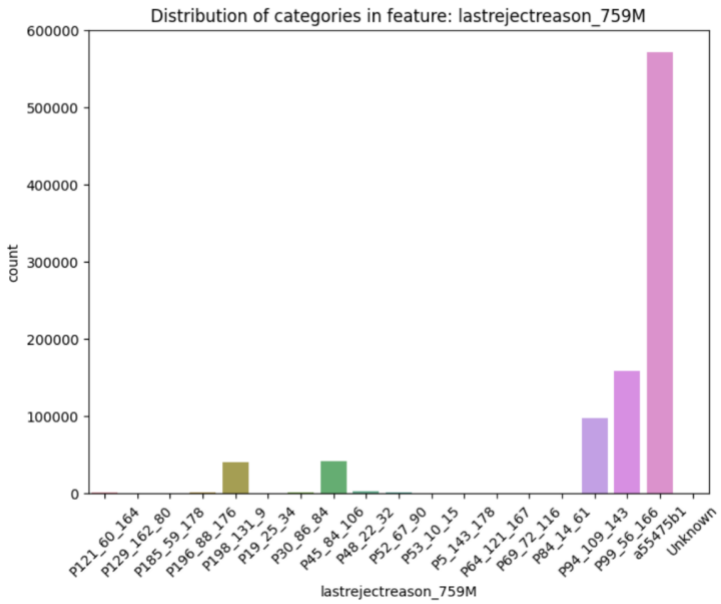


The bar graph labelled "Distribution of categories in feature: lastrejectreason_759M" in Figure 10 depicts the frequency of various rejection reasons in a financial dataset. Each bar indicates a different rejection code, ranging from "P1.60_1.64" to "P9.5_514.55". Most categories

have low counts, indicating that these explanations are less common. However, the "UNKNOWN" category, denoted by a magenta bar, dominates, showing that many rejections are not classified for specific recognized reasons.

Figure 10

Distribution of categories in feature: lastrejectreason_759M

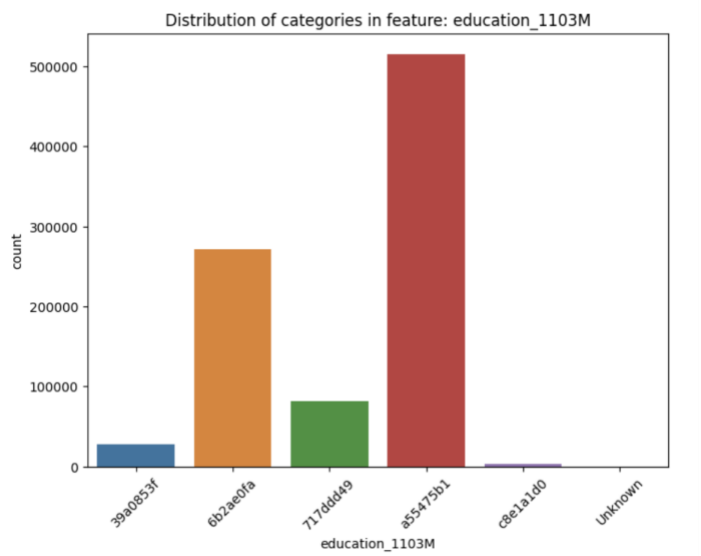


The bar chart in Figure 11 named "Distribution of Categories in Feature: Education_1103M" depicts the frequency of various educational qualifications within a dataset. The categories, represented by codes such as "39a853f", "6b2ae0a", "717dd49", "553751b", "ce2a1d0", and "Unknown", vary greatly in frequency. The "553751b" category, shown by a red bar, is the most common, indicating that this educational status is the most common among the people in the sample. The "Unknown" category, shown by a purple bar, also has a significant count, indicating a large number of entries with unidentified educational content. The other groups emerge much less frequently.

The bar chart in figure 12 depicts the frequency distribution of various marital statuses in a dataset. The categories, denoted by codes like "3439093", "386c01e", "553751b", "a71c6e5", "bc6abe76", "ecd8504", and "Unknown", have different counts.

Figure 11

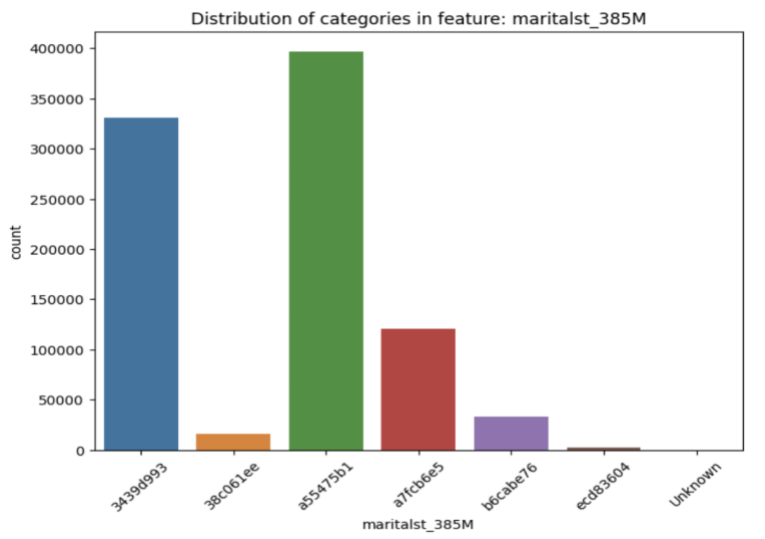
Distribution of categories in feature: education_1103M



The categories "3439093" and "386c01e" are the most common, represented in blue and green, respectively, implying that these marital statuses are the most common in the population analyzed. The "Unknown" category is especially striking, showing a large number of records with an undefined marital status. Other categories, while present, exhibit lesser rates.

Figure 12

Distribution of categories in feature: maritalst_385M



Initially, the dataset containing loan transactions, credit history, and various information about the client in different file formats, such as CSV and Parquet files, was loaded and explored using efficient data-handling libraries to ensure that the tables were loaded properly and

standardized. For example, demographic and financial features are present in base tables, while additional granularity is provided by the static, behavioral, and credit features.

4.1 Data Cleaning and Standardization

To eliminate inconsistencies in the column data types across tables, standardizing the data types for accurate analysis was crucial. For feature encoding, in subsequent steps, numerical columns were converted to appropriate floating-point types, while categorical columns were converted to string or categorical types. This conversion was important to facilitate encoding.

4.2 Feature Engineering

We've used aggregation functions to create meaningful features from grouped tables since the dataset is structured hierarchically. For instance, to provide insights related to the occupation, income sources, and housing types of the client, tables like `train_person_1` and `test_person_1` were aggregated by `case_id`. The `train_credit_bureau_b_2` table was aggregated to identify the maximum overdue payments and late payment values in similar fashion.

4.3 Feature Selection

Relevant features were selected based on domain knowledge and the dataset's documentation. Static features such as `annuity`, `avg_installment_last_24m`, 'credit_amount', and static credit bureau features such as `pmt_average`, `pmt_sum`, and `education_level` were all retained because of their significance in credit risk protection and to enrich the dataset. This type of feature selection process played an important role in reducing dimensionality and improving model performance.

4.4 Data Preparation

To ensure a balanced distribution of positive and negative samples across all sets, tables were merged based on a common `case_id` to consolidate various features into useful training and testing datasets. The `train_basetable` was then joined with static and credit bureau tables to ensure that each `case_id` had all relevant features. A stratified approach was followed to split the consolidated data into training, validation, and testing sets.

4.5 Handling Missing Values

As many features contained missing or unknown data points, handling missing values was an important step. Since it's critical to ensure the features remain usable without introducing biases due to imputed values, our strategy was to fill numeric columns with zero, categorical columns with "Missing", and other data types with "Unknown".

4.6 Encoding Categorical Variables

To represent all the categorical values consistently across all data splits, label encoding was employed to convert categorical features into numeric representations suitable for the models. Each categorical column was encoded using a consistent label encoder trained on the combined data from the training, validation, and test sets.

Finally, categorical features and numerical features were combined into feature arrays suitable for model training. Stacking the encoded categorical and numerical features horizontally helped in creating dense arrays for training, validation, and testing, ensuring that the data was in an appropriate format for subsequent model training. The data preprocessing phase, which involves careful handling of missing values, standardizing data types, aggregating features, and encoding categorical data, made the final dataset consistent and ready for modelling. Each preprocessing step contributed to developing a reliable and accurate predictive model that could assess potential clients' default risk and was also crucial for building a robust predictive model for the Home Credit Stability project.

5. Model Selection

5.1 Baseline Neural Network Model

Model Description: The baseline model is constructed using PyTorch. The model architecture is designed as a feedforward neural network (NN) that processes structured data to predict the likelihood of loan default.

Model Architecture: The neural network comprises several layers:

- Input Layer: Accepts the number of features from the data.
- Hidden Layers: Includes three hidden layers with 128, 64, and 32 neurons, respectively. Each layer uses a ReLU activation function to introduce non-linearity, batch normalization to stabilize the learning process, and dropout with a rate of 0.3 to prevent overfitting.
- Output Layer: A single neuron with a sigmoid activation function to output a probability indicating the likelihood of default.

Hyperparameters:

- Learning Rate: 0.001
- Batch Size: 128

- Number of Epochs: 10
- Optimizer: Adam, which is commonly used for its adaptive learning rate capabilities.
- Loss Function: Binary Cross-Entropy Loss, suitable for binary classification tasks.

Activation Functions: ReLU (Rectified Linear Unit) is used in the hidden layers for non-linear transformations. The output layer uses the sigmoid function to map the output between 0 and 1, representing the probability of a positive class.

Training and Validation:

- The training involves feeding batches of data into the model, where it learns by adjusting weights to minimize the loss. After each epoch, the model's performance is validated using a separate dataset.
- Performance metrics include the area under the receiver operating characteristic curve (AUROC), which provides a single measure of overall accuracy regardless of the classification threshold.

5.2 Deep Classification Network (DCN)

Model Description: A Deep Classification Network (DCN) was developed using the PyTorch deep learning framework for binary classification. The architecture of the DCN comprises densely connected layers and leverages normalization and regularization techniques to minimize overfitting. It was designed to provide accurate predictions and maintain stability across different datasets.

Model Architecture: The Deep Classification Network is built with multiple hidden layers incorporating advanced techniques for optimal performance. The architecture includes the following components:

- Input Layer: The input layer has several features matching the dimensionality of the dataset.
- Hidden Layers: The hidden layers progressively reduce in size, starting at 512 units and ending at 32 units. Each layer applies batch normalization to standardize activations, LeakyReLU activation with a negative slope of 0.1 for non-linearity, and a dropout layer set at 0.25 to ensure optimal data processing while minimizing overfitting.
- Output Layer: To predict the probability of the binary target class, a single output unit is included in the final layer, using the sigmoid activation function.

Hyperparameters: To optimize the performance of the Deep Classification Network, the following hyperparameters were used:

- Optimizer: The Adam optimizer was configured with a learning rate of `0.0001` and a weight decay parameter of `1e-4` to incorporate regularization. It was chosen due to its adaptive learning capabilities.
- Loss Function: Binary Cross-Entropy Loss (BCELoss) was selected as it is well-suited for binary classification tasks.
- Learning Rate Scheduler: The StepLR scheduler was configured with a step size of 10 epochs and a gamma value of 0.1, reducing the learning rate by a factor of 10 every 10 epochs, and the learning rate scheduler was used to dynamically adjust the learning rate during training.
- Epochs: The model was trained for 20 epochs to ensure convergence.
- Batch Size: A batch size of 32 was used to optimize training speed and generalization.

Training and Validation: The Deep Classification Network was trained for 20 epochs, and its performance was evaluated on both training and validation datasets. The loss function and AUROC (Area Under the Receiver Operating Characteristic) metric were used to monitor training progress.

5.3 Deep Neural Network with Residual Block

The Custom Deep Neural Network integrated with Residual Block was designed to strike a balance between model simplicity and predictive performance for numerical or tabular data. The network also leverages residual connections to enable efficient information flow and simplifies the architecture to reduce computational complexity.

Model Architecture: The architecture consists of three primary components:

- Initial Layer: The initial layer reduces the dimensionality of the input features by projecting them into a space of 256 units. Batch normalization is applied to standardize activations, and the LeakyReLU activation function introduces non-linearity.
- Residual Stack:
    - First Residual Block: The first residual block maps the input features from 256 units to 128 units using a linear transformation. Batch normalization and LeakyReLU activation function are employed within the block to maintain activation stability

and non-linearity. A shortcut connection is used to ensure efficient information flow.

- Second Residual Block: Similarly, the second residual block projects features from 128 units down to 64 units while maintaining the residual connection pattern.

- The final layers comprise a linear transformation from 64 units to 16 units, followed by a LeakyReLU activation function. The output layer includes a linear transformation to a single unit with a sigmoid activation function to predict the binary target class.

Hyperparameters:

- Optimizer: The Adam optimizer configured with a learning rate of 0.001 and a weight decay parameter of 1e-4 was selected for its adaptive learning capabilities,
- Loss Function: Binary Cross-Entropy Loss (BCELoss) was used due to its suitability for binary classification tasks,
- Learning Rate Scheduler: A cosine annealing learning rate scheduler (CosineAnnealingLR) configured with a maximum learning period (T_max) was used to gradually reduce the learning rate,
- Epochs: The model was trained for 8 epochs to ensure convergence.
- Batch Size: A batch size of 32 was used to optimize training speed and generalization.

5.4 Deep Neural Network with VGGnet-inspired block

Model Description: The deep neural network model is inspired by the VGG architecture and adapted for tabular data. This model is designed to handle structured data with a fixed number of features, processing it through a series of linear layers and nonlinear activations to perform binary classification.

Model Architecture: To effectively learn and predict structured tabular data, the model architecture makes use of numerous layers of linear transformations, activations, and normalizations. The use of ReLU and batch normalization aids in the stabilization of the training process, while the dropout minimizes overfitting by adding regularization. This architecture works especially well for datasets with complicated and non-linear feature relationships.

- Initial Layer: It maps the input feature dimensions to a higher-dimensional space of 256 units.
- Hidden layer: The VGGNetTabular model features a series of hidden layers organized into three main blocks, designed to progressively transform and refine the input features:

    - Block 1 consists of two sublayers, each with a linear transformation that keeps the feature dimension at 256 units. The ReLU activation function introduces nonlinearity, and the activations are stabilized by batch normalization. Dropout is used at a rate of 0.1 to help prevent overfitting.
    - Block 2 reduces the feature dimensions from 256 to 128 units over two sublayers. Each sub-layer features ReLU activation and batch normalization, as well as dropout to reduce the risk of overfitting.
    - Block 3 reduces the feature dimensions from 128 to 64 units, following the same structure as the previous blocks: linear layers, ReLU activation, batch normalization, and dropout.
- Output layer: The output layer employs a sigmoid function to provide a probabilistic interpretation, making the model appropriate for binary classification problems.

Hyperparameters:

- Optimizer: The Adam optimizer was chosen due to its adaptive learning characteristics, which aid in efficiently converging to the optimal solution. The optimizer is configured with a learning rate of 0.001 and a weight decay of 1×10^{-4}. This regularization parameter penalizes big weights, hence preventing overfitting.
- Loss of Function: Due to its applicability for binary classification tasks, the loss function used is Binary Cross-Entropy Loss (BCELoss). This loss function calculates the difference between the predicted probabilities and the actual binary labels.
- Learning Rate Scheduler: To gradually reduce the learning rate, a cosine annealing learning rate scheduler (CosineAnnealingLR) was utilized, with a maximum learning period of 50 epochs set.
- Epochs: The model was trained for eight epochs to ensure convergence.
- Batch Size: A batch size of 32 was used to maximize training speed and generalization.

5.5 *BiLSTM with SMOTE*

Model Description:

This model is constructed using PyTorch and is designed as a Bidirectional Long Short-Term Memory (BiLSTM) network aiming to predict the likelihood of clients defaulting on loans by analyzing the Home Credit dataset.

Data Preprocessing:

- SMOTE: Given the class imbalance typically present in default prediction datasets, Synthetic Minority Over-sampling Technique (SMOTE) is employed to artificially balance the dataset by generating synthetic samples, ensuring robust model training.
- Feature Engineering: Extensive feature engineering is applied to derive meaningful attributes from raw data, including aggregation and transformation techniques.

Model Architecture:

- Input Layer: The model accepts input data reshaped and preprocessed to fit into a sequence format suitable for time-step analysis, emphasizing the sequential nature of financial data.
- BiLSTM Layer: The core of the model is a Bidirectional Long Short-Term Memory (BiLSTM) layer, designed to capture patterns from both forward and reverse directions of the data sequence, thereby providing a comprehensive understanding of temporal dynamics.
- Dense Layer: After the LSTM layers, the network includes a fully connected dense layer to interpret the LSTM outputs, leading to the final prediction.
- Output Layer: Utilizes a sigmoid activation function to output a probability, indicating the likelihood of a loan default.

Hyperparameters:

- Learning Rate: 0.0001
- Batch Size: 128
- Epochs: 20, with early stopping based on validation loss to prevent overfitting.
- Optimizer: Adam optimizer with weight decay to stabilize the learning process.
- Loss Function: Binary Cross-Entropy

Training and Validation:

- The model is trained using a batch-wise approach with data loaders, allowing efficient handling of large datasets.
- Performance is monitored using the AUROC metric, providing a comprehensive measure of model accuracy across different thresholds.
- Implements early stopping based on validation performance to halt training when improvements cease, preserving model generalization.

6. Evaluation, Result Analysis and Visualizations

6.1 *Baseline Neural Network Model*

The baseline deep learning model has been evaluated using several key metrics to assess its performance in predicting loan defaults. The primary metrics include Area Under the Receiver Operating Characteristic Curve (AUROC), Gini Stability, and Binary Cross-Entropy Loss. This evaluation report summarizes the findings from these metrics based on the model's application to training, validation, and test datasets.

AUROC (Area Under the Receiver Operating Characteristic Curve): (Refer to Figure 13)

- Training Set: The model demonstrated a progressive improvement in AUROC over ten epochs, indicating effective learning and adaptation to the training data. The AUROC started at approximately 0.6329 and reached 0.7099 by the final epoch.
- Validation Set: The AUROC on the validation set showed a consistent increase alongside the training trend, peaking at 0.7194. This suggests that the model was not only fitting to the training data but was also generalizing well to unseen data.
- Test Set: The final evaluation of the test set yielded an AUROC of 0.7163. This score is slightly lower than the validation set but still indicates a good predictive ability on entirely new data.

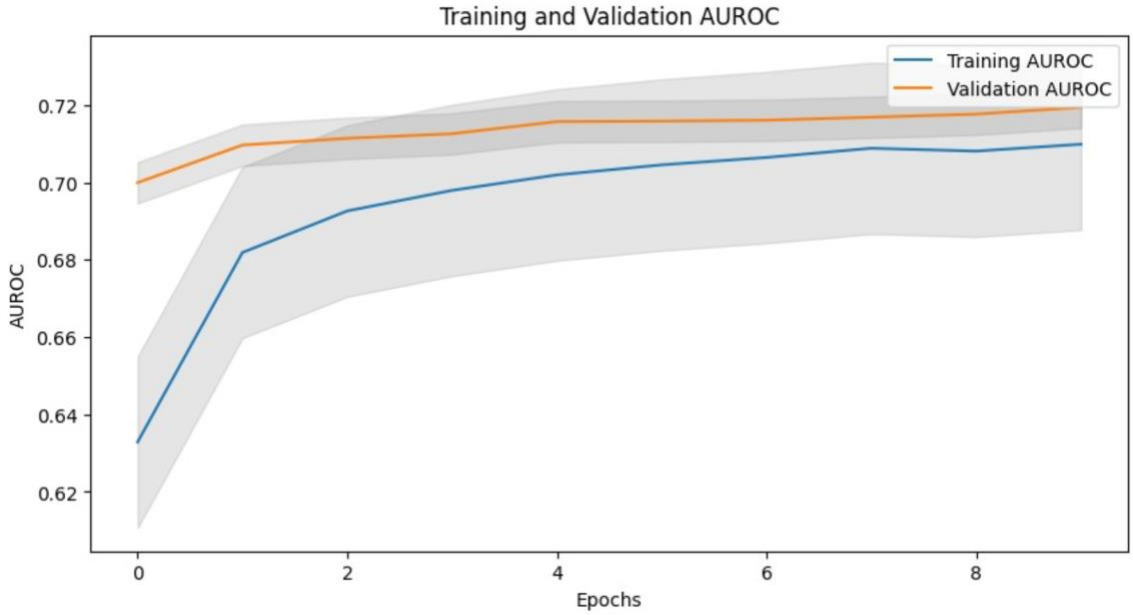
Gini Stability:

- Training Set: The model achieved a Gini stability score of 0.4232, reflecting moderate stability in the training data predictions across different time frames.
- Validation Set: A slightly lower stability score of 0.4110 was observed on the validation set, suggesting a small decrease in predictive consistency on unseen data.

- Test Set: The stability score further decreased to 0.3986 on the test set. This decrement indicates that while the model is reasonably stable, variations across different temporal segments affect its performance.

Figure 13

Training and Validation AUC-ROC Scores with Standard Deviation Shading (Baseline Model)



Loss Metrics (Binary Cross-Entropy Loss):

- Training Process: The loss metrics reported during training showed a decreasing trend, starting from an initial average of 0.1490 and reducing to 0.1309 by the end of training. This decrease in loss is consistent with the improvements in the AUROC metric, confirming effective learning.
- Validation Process: The validation loss started lower than the training loss, indicating good initial generalization. However, the loss experienced some fluctuations, indicating potential areas for model improvement to enhance stability and performance consistency. Refer to Figure 14 to visualize the training and validation loss trends.

Figure 14

Training and Validation Losses with Standard Deviation Shading (Baseline Model)

6.2 Deep Classification Network (DCN)

The performance of the Deep Classification Network (DCN) was evaluated by training and validating the model across 20 epochs. The following metrics were monitored throughout the training process:

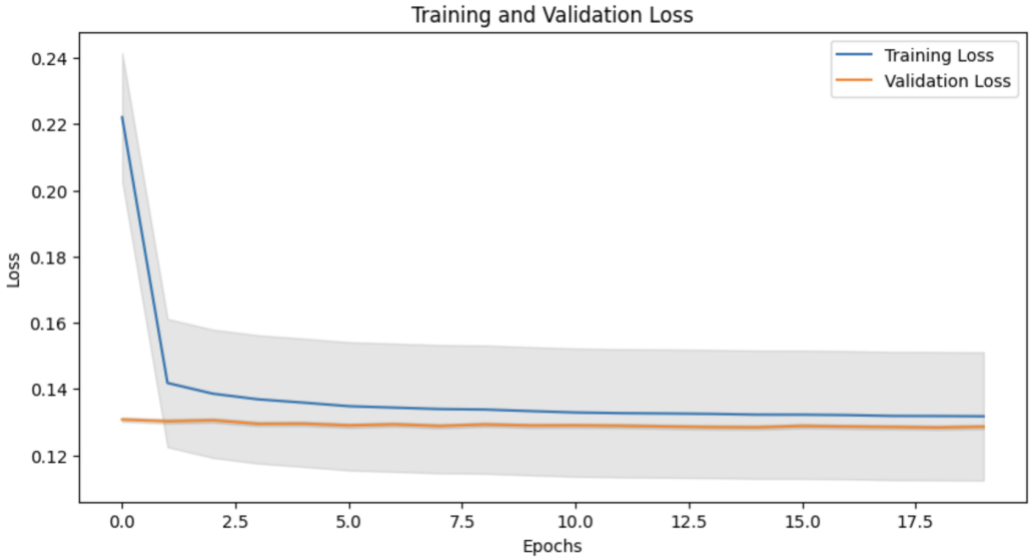
Loss Analysis: The training and validation loss curves over 20 epochs are illustrated in Figure 15. The training loss exhibits a steady decline, whereas the validation loss remains relatively stable, suggesting that the model is learning effectively without significant overfitting.

Stability Score Analysis: The stability score measures the robustness of the model's predictions across different datasets. The stability score achieved by DCN for the training, validation, and test sets is 0.4157497921492467, 0.3924292414286484, and 0.39633677138387113, respectively indicating that the model maintains a reasonable level of consistency in its predictions across various datasets, though there is room for improvement in robustness.

AUROC Analysis: The AUROC (Area Under the Receiver Operating Characteristic) score reflects the model's ability to distinguish between positive and negative classes. AUROC scores achieved by DCN are 0.7210383329301211, 0.7140300604477774, and 0.7148901854877854 for training, validation, and the test set, respectively.

Figure 15

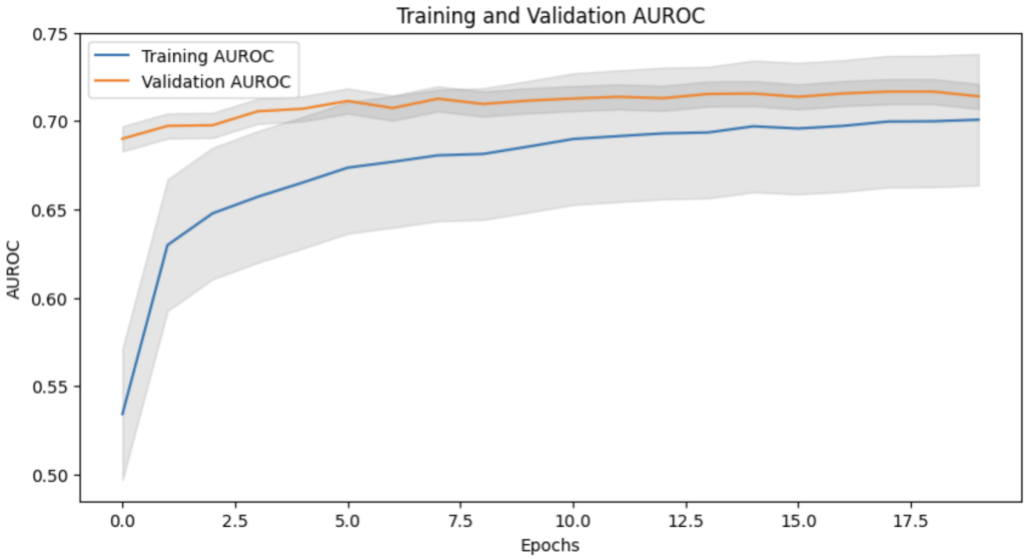
Training and Validation Losses for DCN



These results show that the DCN demonstrates satisfactory performance in terms of predictive accuracy, with consistent scores across the training, validation, and test datasets. The AUROC curves for both training and validation datasets over 20 epochs are presented in Figure 16. The training AUROC shows steady improvement over the epochs, while the validation AUROC remains consistent, reflecting stable performance.

Figure 16

Training and Validation AUROC for DCN



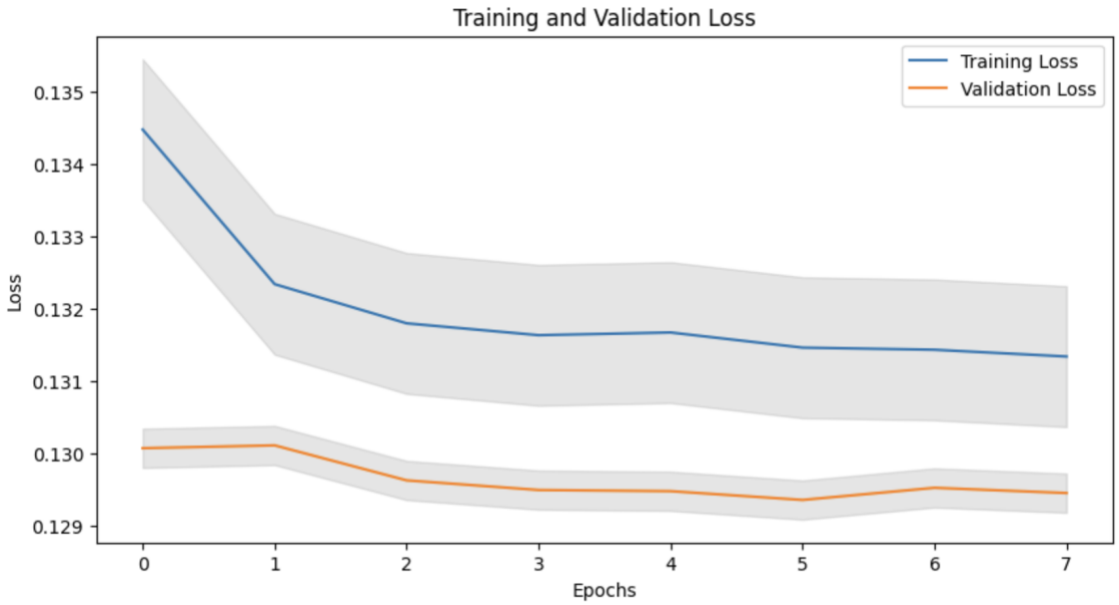
6.3 Deep Neural Network with Residual Block

Loss Analysis: Figure 17 displays the training and validation loss curves over 8 epochs. The training loss decreases steadily, while the validation loss remains relatively constant, indicating that the model is learning effectively without significant overfitting.

Stability Score Analysis: The stability score assesses the consistency of the model's predictions across different datasets. The model achieved stability scores of 0.3978097947642462 for the training set, 0.3797369627585283 for the validation set, and 0.3800129314028726 for the test set. These scores reflect a reasonable degree of consistency in the model's predictions across various datasets.

Figure 17

Training and Validation Losses with Standard Deviation (DNN with Residual Block)

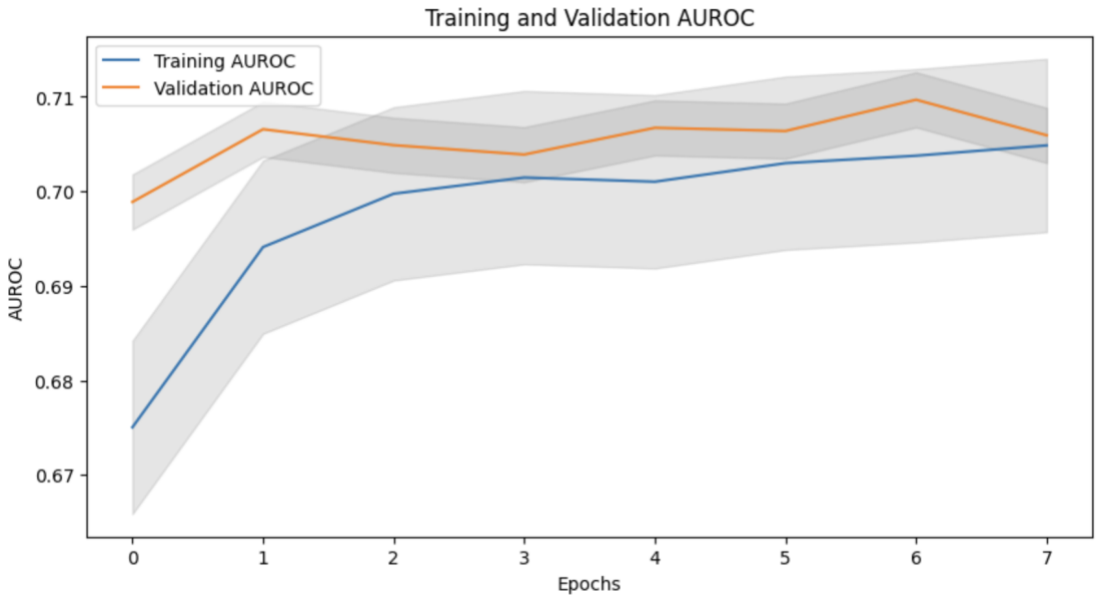


AUROC Analysis: The AUROC (Area Under the Receiver Operating Characteristic) score measures the model's ability to distinguish between positive and negative classes. The Deep Neural Network with residual block achieved AUROC scores of 0.7116795176909101 for the training set, 0.7059289855214705 for the validation set, and 0.7063531815674403 for the test set. These scores indicate moderate predictive accuracy, with consistent performance across the training, validation, and test datasets.

Figure 18 presents the AUROC curves for both training and validation datasets over 8 epochs. The training AUROC score improves consistently throughout the epochs, while the validation AUROC score remains relatively constant, demonstrating stable model performance.

Figure 18

Training and Validation AUROC with Standard Deviation (DNN with Residual Block)



6.4 Deep Neural Network with VGGnet-inspired block

Loss Analysis: Figure 19 shows the training and validation loss curves over eight epochs. The training loss rapidly lowers, but the validation loss remains roughly constant, showing that the model is learning efficiently and without severe overfitting.

The stability score measures the consistency of the model's predictions across multiple datasets. The deep neural network with vggnet block obtained stability scores of 0.38431107705726525 for the training set, 0.3751160517419537 for the validation set, and 0.36911922586531865 for the test set. These ratings indicate a reasonable level of consistency in the model's predictions across multiple datasets.

AUROC Analysis: The AUROC (Area Under the Receiver Operating Characteristic) score assesses the model's ability to discriminate between positive and negative categories. The VGGNET had an AUROC score of 0.7041638925262816 for the training set, 0.7023145766778215 for the validation set, and 0.7007068138196983 for the test set. These results show moderate predictive accuracy and consistent performance throughout the training, validation, and test datasets.

Figure 19

Training and Validation Losses with Standard Deviation (DNN with VGGNET block)

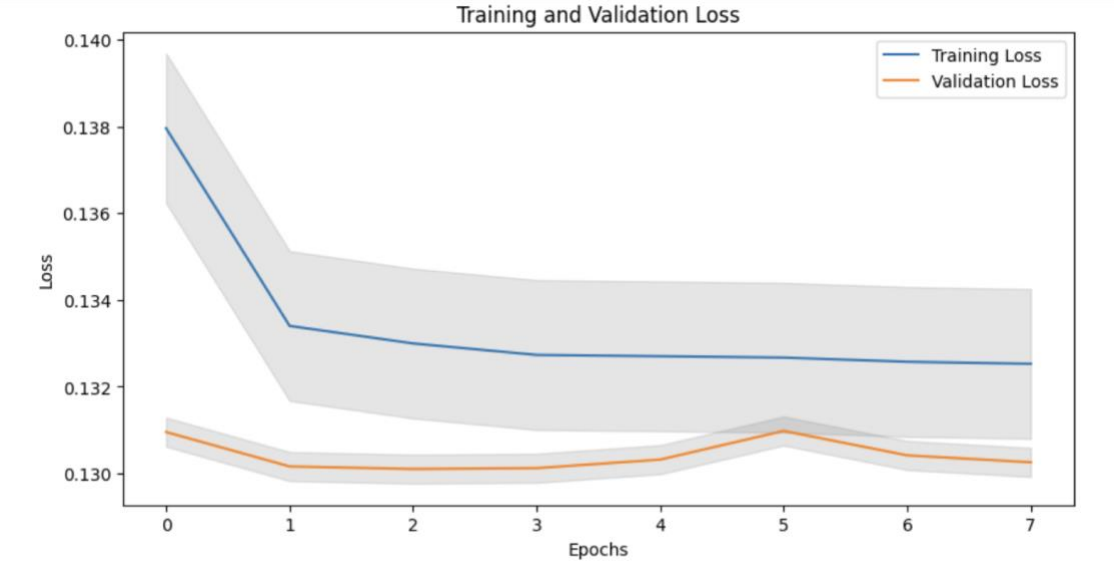
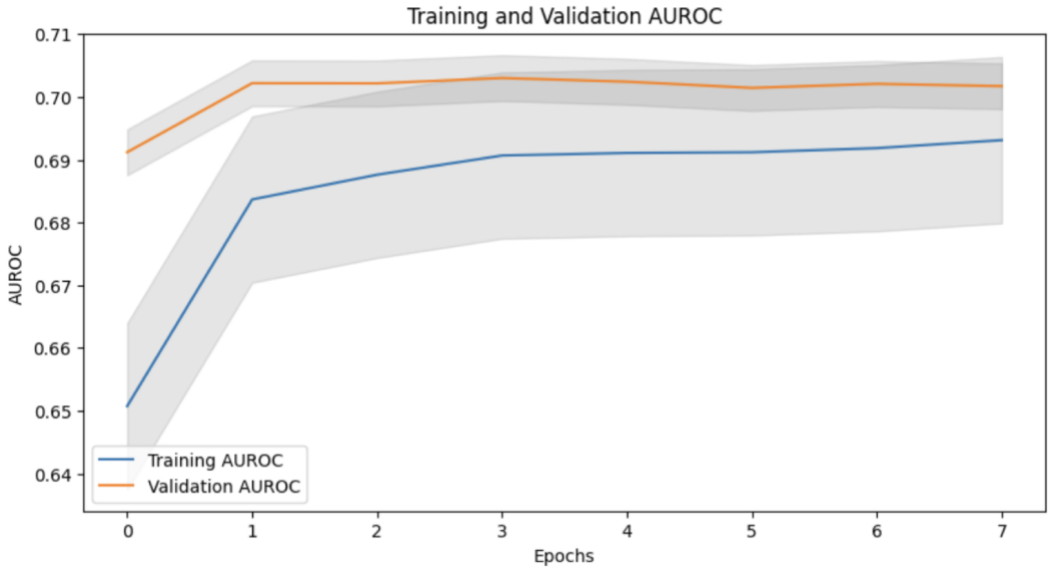


Figure 20 shows the AUROC curves for both the training and validation datasets over eight epochs. The training AUROC score steadily improves during the epochs, however, the validation AUROC score remains relatively constant, indicating steady model performance.

Figure 20

Training and Validation AUC-ROC Scores with Standard Deviation (DNN with VGGNET block)



6.5 BiLSTM with SMOTE

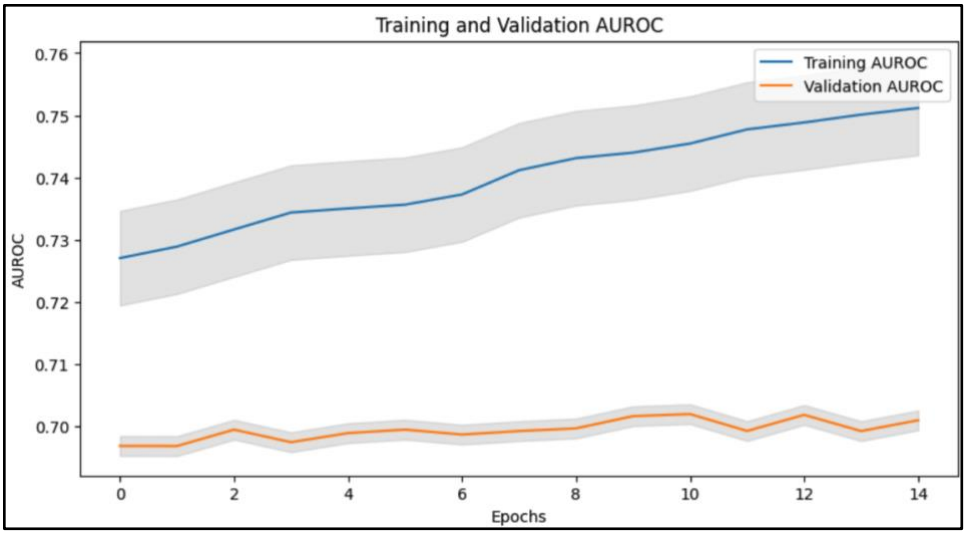
AUROC Analysis

The model demonstrated a consistent performance across different datasets, as evidenced by the Area Under the Receiver Operating Characteristic Curve (AUC) scores: 0.718 on the training set,

0.700 on the validation set, and 0.703 on the test set. These results indicate a good predictive capability, with the model showing slightly better performance during training compared to the validation and testing phases as shown in Figure 21.

Figure 21

Training and Validation AUC-ROC Scores with Standard Deviation (BiLSTM Model)



Gini Stability

The stability scores, which measure how consistent the model's predictions are over time, were 0.4107 for the training set, 0.3687 for the validation set, and 0.3618 for the test set. While there is a slight decrease in stability from training to testing, these scores still reflect a reasonable level of reliability in the model's temporal robustness.

Loss Analysis

During the 20 epochs, the model demonstrated consistent progress, with training loss steadily decreasing from 0.6060 to 0.5864, indicating an improvement in the model's predictive accuracy on the training dataset. On the validation side, the loss exhibited more variability, oscillating between 0.5973 and 0.6407. This fluctuation suggests some variability in model performance when exposed to unseen data. Despite these variations in loss, the overall trend shows that the model is effectively adapting its parameters to capture underlying patterns in the training data, thereby maintaining a good level of generalization across different data sets as illustrated in Figure 22.

Figure 22

Training and Validation Losses with Standard Deviation (BiLSTM Model)

Table 1

Model Evaluation Comparison

Model	***Baseline***	***DNN***	***DNN with Residual Block***	***BiLSTM***	***DNN with VGGNET block***
Train Loss	0.130	0.138	0.132	0.586	0.132
Validation Loss	0.147	0.129	0.130	0.598	0.130
Train AUC	0.724	0.721	0.711	0.718	0.704
Validation AUC	0.719	0.714	0.706	0.700	0.702
Test AUC	0.716	0.715	0.706	0.702	0.700
Stability Score: Train	0.423	0.416	0.398	0.410	0.384
Stability Score: Valid	0.411	0.392	0.380	0.368	0.375
Stability Score: Test	0.398	0.396	0.380	0.361	0.369

Kaggle Submission Public Score	0.347	0.394	0.370	0.320	0.394

Leaderboard Rank on Kaggle

7. Conclusion and Future Work

Conclusion

In our analysis, we evaluated various deep-learning models to predict loan default risks for Home Credit, focusing on individuals with minimal or no credit history. The models compared included a Baseline model, Deep Classification Network (DCN), DNN with Residual Block, Bidirectional Long Short-Term Memory (BiLSTM), and Deep Neural Network with VGGnet-inspired block. These were assessed based on training and validation losses, AUROC scores, and stability metrics.

The Baseline model exhibited the highest overall stability scores (Train: 0.423, Validation: 0.411, Test: 0.398) and maintained consistent AUROC scores across all datasets, demonstrating its robustness and reliability in performance without complex features. The Deep Classification Network (DCN) showcased strong performance, particularly in stability and AUROC scores, suggesting its effectiveness in capturing both explicit and intricate interactions in the data. The DNN with Residual Block displayed decent performance metrics but lagged slightly in stability, indicating potential issues with capturing longer-term dependencies or overfitting to the training data. The Bidirectional LSTM (BiLSTM), despite its higher training and validation losses, showed competitive AUROC scores (Train: 0.718, Validation: 0.700, Test: 0.702) and capitalized on its ability to process sequences with context from both past and future data points, offering a

significant advantage in understanding temporal dependencies crucial for accurate predictions. Lastly, Deep Neural Network with a VGGnet-inspired block was adapted for sequence data, showing moderate success but highlighting the challenges of applying convolutional network architectures directly to sequence prediction tasks.

Each model has its strengths- the Baseline for its simplicity and stability, DCN for handling feature interactions, BiLSTM for enhanced temporal analysis, and Deep Neural Network with VGGnet-inspired block for experimenting with convolutional approaches in sequential tasks.

Future Work

To further enhance the model's effectiveness and address the observed limitations, we propose the following areas for future research:

1. Interpretable AI: Advancing techniques that increase the interpretability of deep learning models will help align them with regulatory expectations. Tools such as LIME or SHAP could elucidate how neural networks make decisions, increasing their transparency.
2. Model Stability: Given the fluctuating nature of consumer financial behaviour, improving the stability of predictive models over time without compromising their accuracy is essential.
3. Hybrid Models: Merging the predictive power of deep learning with the transparency of traditional models could result in a hybrid approach that might be more readily accepted by industry practitioners and regulators.
4. Data Augmentation: Beyond SMOTE, exploring other data augmentation strategies like adversarial training or synthetic data generation could potentially reduce overfitting and enhance model generalization.
5. Advanced Architectures: Experimenting with more sophisticated neural network designs could further improve the capability of models to process and predict based on complex financial data patterns.

References

1. Gaspar, Peter, Mwananchipeta, Mwembezi., N., Kalimang’asi., Geoffrey, A, Lusanjala. (2022). Determinants of loan defaults in two selected financial institutions in Sumbawanga municipality, Tanzania. American journal of finance, 7(1):48-60. doi: 10.47672/ajf.1000

2. Ebenezer, Owusu., Richard, Quainoo., Justice, Kwame, Appati., Solomon, Kuuku, Mensah. (2022). Loan Default Predictive Analytics. 617-622. doi: 10.1109/AIC55036.2022.9848906
3. Muhamad, Abdul, Aziz, Muhamad, Saleh, Jumaa., Mohammed, Saqib. (2023). Improving Credit Risk Assessment through Deep Learning-based Consumer Loan Default Prediction Model. International Journal of Finance & Banking Studies, 12(1):85-92. doi: 10.20525/ijfbs.v12i1.2579
4. Shasha, Liu., Ming, Shan, Guan., Yang, Li., Menglu, Wang., HuiMin, Zhu. (2023). A Bayesian deep learning method based on loan default rate detection. 12635:1263516-1263516. doi: 10.1117/12.2678879
5. (2023). Multi-view GCN for Loan Default Risk Prediction. doi: 10.21203/rs.3.rs-2754272/v1
6. Ebenezer, Owusu., Richard, Quainoo., Solomon, Kuuku, Mensah., Justice, Kwame, Appati. (2023). A Deep Learning Approach for Loan Default Prediction Using Imbalanced Dataset. International Journal of Intelligent Information Technologies, 19(1):1-16. doi: 10.4018/ijiit.318672
7. Yanzhen, Qu., Ihsan, Said. (2023). Improving the Performance of Loan Risk Prediction based on Machine Learning via Applying Deep Neural Networks. European Journal of Electrical Engineering and Computer Science, 7(1):31-37. doi: 10.24018/ejece.2023.7.1.475
8. Adaleta, Gicić., Dženana, Đonko. (2023). Proposal of a model for credit risk prediction based on deep learning methods and SMOTE techniques for imbalanced datasets. 1-6. doi: 10.1109/ICAT57854.2023.10171259
9. Shivaram, Hegde. (2023). UQ for Credit Risk Management: A deep evidence regression approach. doi: 10.48550/arxiv.2305.04967
10. Xin Huang, Ashish Khetan, Milan Cvitkovic, Zohar Karnin. (2020). TabTransformer: Tabular Data Modeling Using Contextual Embeddings. doi: https://arxiv.org/abs/2012.06678