



ALBUKHARY INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTING AND INFORMATICS

Project: Deep Neural Network

Course Code	CCS3113	Course Name	Deep Learning
Group Members	Thinley Yeshey Choden		AIU22102188
	Malika Amiri		AIU21102298
	Dema Yuden		AIU22102234

Assessment Marks:

No.	Criteria	Weight	Marks
1			
2			
	Total Marks		

Comments:

--

Date of Submission: 20/01/2025

Lecturer's Name: Dr. Mozaherul Hoque

Table of contents

1.0 Introduction	3
1.1 Problem Statement	3
1.2 Importance and Relevance	3
1.3 Prior Work and Existing Solutions	3
1.4 Proposed Solution	4
2.0 Methods	5
2.1 Dataset Overview	5
2.1.1 Dataset Features	5
2.1.2 Description of variables in train/test.csv	5
2.1.3 Targets	6
2.2 Data Preprocessing	6
2.3 Model Selection	7
2.4 Model Architecture	7
2.5 Training Procedure	8
2.6 Evaluation Metrics	9
2.7 Predictions and Final Model Evaluation	9
3.0 List of Efforts to Improve Ranking	10
3.1 Overview of Efforts	10
3.2 Sequential Batch Refinements and Experiments in Model Optimization	13
3.2.1 Baseline Model	13
3.2.2 Data Preprocessing and Feature Engineering	13
3.2.3 Outlier Handling and Column Reduction	13
3.2.4 Model Optimization and Regularization	13
3.2.5 Neural Network Architecture Tuning	14
3.3 Result Visualization and Interpretation	15
3.4 Final Ranking	17
4.0 Conclusion / Future work	17

1.0 Introduction

The African Credit Scoring Challenge by Zindi is one of the most ambitious undertakings aimed at enhancing the robustness of credit risk analysis, a very significant part of financial decision-making in the rich and fast-developing African economy. Participants in this competition are supposed to predict loan repayments from customer profiles by using anonymized data with machine learning, deep learning, and applied deep learning for maximum financial efficiency.

1.1 Problem Statement

The problem relates to loan defaults, which is a critical problem faced by every financial institution in the world, and even more so in the dynamic and diverse financial markets of Africa. Correctly assessing the probability of loan defaults helps financial institutions smooth out risks and optimize lending decisions. Economic dynamism, demographic diversity among customers, and inequality in credit history, make this task more challenging yet crucial for financial stability and growth in Africa.

1.2 Importance and Relevance

The efficiency of loan default prediction would greatly affect the long-term viability of financial institutions. Efficiently predicted loan defaults bring minimal financial losses, manageable risk exposure, and the development of sustainable lending practices. Besides, better risk assessment would facilitate the possibility of reaching more unserved or underserved people by financial institutions, increasing financial inclusion across the continent.

1.3 Prior Work and Existing Solutions

Prior work in machine learning for credit scoring has highlighted challenges such as flawed data, uncertain regulations, and data security concerns. Models often struggle with biased predictions due to incomplete datasets that exclude credit-thin consumers, and regulatory uncertainty creates hesitancy in adoption. Additionally, data security remains a critical issue, as sophisticated models can inadvertently reveal sensitive characteristics. Despite these challenges, machine learning

continues to show promise in improving credit assessments, as demonstrated by digital lenders and financial institutions exploring new methods to enhance accuracy and fairness.

1.4 Proposed Solution

The objective of this project is to explore the domain of deep learning techniques, specifically the Feedforward Neural Network (FNN) to make a prediction on the problem of loan defaults in the diverse financial markets of Africa. We adopt the deep learning technique in order to enhance the accuracy of the model so that it would be better positioned to address specific challenges brought forth by diversity in financial markets in Africa.

2.0 Methods

2.1 Dataset Overview

The **African Credit Scoring Challenge** by Zindi provides a dataset that can be used as input for the task of predicting probabilities of loan defaults under different financial and economic contexts. The dataset provides loan and customer information from Kenya for the training and test sets and for Ghana, a test set only, thus allowing the development of model that generalize well across countries. Below are its key characteristics:

2.1.1 Dataset Features

Main datasets:

- **Train.csv:** This file contains the data representing a training set, including both the features and the target variable. It is to be used in training a DNN model on it that can predict the target variable for the test set.
- **Test.csv:** It contains the exact same structure as the train.csv but without the target column.

Additional dataset:

- **economic_indicators.csv:** This dataset contains additional economic data from the Federal Reserve Economic Data (FRED) portal in order to add external features to enhance the prediction model.
- These economic indicators provide context to the loan data and might have an impact on the probability of loan defaults, enriching the feature set of the model.

2.1.2 Description of variables in train/test.csv

0. **ID:** A unique identifier for each entry in the dataset.
1. **customer_id:** Unique identifier for each customer in the dataset.
2. **country_id:** Identifier or code representing the country where the customer resides or where the loan was issued.

3. **tbl_loan_id:** Unique identifier for each loan associated with the customer.
4. **Total_Amount:** The total loan amount initially disbursed to the customer.
5. **Total_Amount_to_Repay:** The total amount the customer is expected to repay, including principal, interest, and fees.
6. **loan_type:** The category or type of loan.
7. **disbursement_date:** The date when the loan amount was disbursed to the customer.
8. **duration:** The length of the loan term, typically expressed in days
9. **lender_id:** Unique identifier for the lender or institution that issued the loan.
10. **New_versus_Repeat:** Indicates whether the loan is the customer's first loan ("New") or if the customer has taken loans before ("Repeat").
11. **Amount_Funded_By_Lender:** The portion of the loan funded directly by the lender.
12. **Lender_portion_Funded:** Percentage of the total loan amount funded by the lender.
13. **due_date:** The date by which the loan repayment is due.
14. **Lender_portion_to_be_repaid:** The portion of the outstanding loan that needs to be repaid to the lender.
15. **target:** This variable takes the value 0 or 1. 1 means the customer defaulted on the loan, whereas 0 means, the customer paid the loan.

2.1.3 Targets

- **Target Variable:** The `train.csv` includes a binary target where 0 represents that loan was successfully repaid and 1 represents a loan default.

2.2 Data Preprocessing

1. **Removal of Irrelevant Features:** Features like ID, `tbl_loan_id`, `customer_id`, `country_id` were dropped while training the model as it didn't contribute significantly to models training.
2. **Feature Transformation:** `country_id`, `New_versus_Repeat` consisted of categorical variables which was encoded using label encoding. `Loan_type` consisted of categories of different loans which was encoded using one-hot encoding.
3. **Feature Scaling:** Numerical columns like `Total_Amount`, `Total_Amount_to_Repay`, `Amount_Funded_By_Lender`, `Lender_portion_Funded`,

Lender_portion_to_be_repaid were scaled using Standard Scaling to normalize the data before training.

4. **Handling Outliers:** The numerical columns consisted of outliers which were detected and handled using the IQR method.
5. **Date Parsing:** **disbursement_date** and **due_date** were parsed into separate week, month, and year columns to better capture any temporal patterns.
6. **Handling Data Imbalance:** The dataset is quite imbalanced, with more non defaults existing than actual defaults. This has been balanced using SMOTE: Synthetic Minority Over-sampling Technique. SMOTE interpolates between instances to create synthetic examples of the minority class (loan defaults). It reduces the bias of the classifier for the majority class and improves model performance.

2.3 Model Selection

The Feedforward Neural Network (FNN) was chosen for its simplicity and effectiveness in handling the structured data related to the binary classification problem of loan defaults. Normally, FNNs are applied to problems of this nature, since it is possible to understand complex relationships between features in a multilayer structure, capturing the linearity and non-linearity underlying the data.

Besides, FNNs are computationally efficient and allow the tuning of depth and complexity of the neural network in order to optimize model performance. This makes them feasible for the problem at hand, where numerical and categorical features each take their turn influencing the target variable.

2.4 Model Architecture

1. Feed-Forward Neural Network:

- The model architecture consists of three fully connected hidden layers:
 - **First Hidden Layer:** 128 neurons with L2 regularization, batch normalization, and LeakyReLU activation.
 - **Second Hidden Layer:** 64 neurons with L2 regularization, batch normalization, and LeakyReLU activation.

- **Third Hidden Layer:** 32 neurons with LeakyReLU activation.
- **Output Layer:** A single neuron with sigmoid activation, as this is a binary classification problem.

2. Regularization:

- We prevented overfitting by utilizing **Dropout** with a rate of 0.3 in each hidden layer. In Dropout, a fraction of neurons is dropped stochastically during the training process for better generalization.
- Also, **L2 regularization** is added in the first two layers, which will penalize the large weights and avoid overfitting.

3. Class Weight Adjustment:

- **Class weights** were calculated as another approach toward solving this problem of class imbalance. These weights are given to the model while training to make sure that it appropriately emphasizes the minority class while learning.

2.5 Training Procedure

1. Optimizer and Loss Function:

- The model was then compiled with the **Adam optimizer** and the **binary cross-entropy** loss function. Adam will work pretty well for such a task; it changes the learning rate while training and provides efficient optimization.

2. Learning Rate Scheduler:

- To adapt learning rates at different parts of the training process, a learning rate scheduler was implemented. The initial learning rate is 0.001, which, with a decay rate of 0.5 for every 10th epoch, will make the updates smaller and smaller as convergence is reached. The total epochs was 50.

3. Early Stopping:

- Applied **early stopping**, which monitors the validation loss to prevent the model from continuing training when there is no improvement in the model's performance in order to prevent overfitting. The early stopping method restores the weights of the best model.

2.6 Evaluation Metrics

- **Train-Test Split:** The data was divided into an 80-20 ratio for training versus validation, respectively, in order to check the performance of the model at the time of training.
- The model is evaluated on the validation set using accuracy and **F1 score (evaluation metric for this competition)**. The relevant scores will help in assessing the balance between precision and recall, which is suitable for such an imbalanced nature of the target variable.

2.7 Predictions and Final Model Evaluation

After training the model, we load it and use it to predict the target values for the test dataset, which does not include the target column. Predicted output is binary 0 or 1, based on threshold value 0.5.

Prediction Output: Finally, predicted target variables forecasted by the model were appended to the test.csv dataset for submission wherein every test sample has a prediction which will be scored on the Zindi platform.

3.0 List of Efforts to Improve Ranking

3.1 Overview of Efforts

A total of 24 submissions were made by our team out of which 21 correct submissions are discussed in **Table 1**, along with their public (approximately 30% of the test dataset) and private scores (70% of the test dataset). The following list of efforts were made to increase our F1 score and improve our rank in the leaderboard.

Table 1. Timeline of Submissions

No.	Date	Changes Implemented	Public Score	Private Score	Ranking
1	31/12/2024	Used the starter notebook as the basis to implement a simple feed forward neural network.	0.232380952	0.238881829	621
2	02/01/2025	Tried MinMax Scaling.	0.139999999	0.141680395	626
3	02/01/2025	Encoding categorical columns (One-Hot encoding) to include categorical features in model prediction.	0.438709677	0.435058078	576
4	03/01/2025	Dates were dissolved into separate columns to convert the column into understandable and correct format.	0	0	584
5	03/01/2025	Oversampled the lesser class group to address class imbalance and ensure minority class (target 1) is	0.03649635	0.077108433	584

		adequately represented.			
6	03/01/2025	Included additional features to provide the model with more relevant information, helping it to improve its ability to capture complex patterns.	0.287841191	0.300713985	584
7	05/01/2025	Outlier handled from numerical columns using IQR method to reduce the impact of outliers by converting closer values with the rest.	0.248366013	0.236518448	603
8	05/01/2025	Dropped Irrelevant ID columns	0.259030837	0.241969378	603
9	05/01/2025	Implemented early stopping with a patience rate of 5.	0.248366013	0.236518448	603
10	11/01/2025	Implemented early stopping with a patience rate of 10.	0.040797824	0.040653466	669
11	12/01/2025	Added learning rate scheduling function with initial learning rate of 0.001	0.010752688	0.003430531	682
12	13/01/2025	Used Leaky ReLU	0.070934256	0.07199084	693
13	13/01/2025	Added L2 Regularization.	0.290456431	0.305364511	673

14	13/01/2025	Implemented batch normalization	0.3671875	0.380187416	662
15	13/01/2025	Increased to 50 epochs.	0.448717948	0.466737064	631
16	13/01/2025	Increased the number of neurons per layer to 128, to capture more complex patterns.	0.413114754	0.402597402	631
17	13/01/2025	Experimented with drop out rates 0.2.	0.456591639	0.455508474	631
18	13/01/2025	Experimented with drop out rates 0.4.	0.333333333	0.343634116	631
19	13/01/2025	Experimented with drop out rates 0.3.	0.406666666	0.413867822	631
20	13/01/2025	Increased to 100 epochs.	0.445859872	0.464921465	631
21	13/01/2025	Increased to 200 epochs.	0.402684563	0.413186813	631

3.2 Sequential Batch Refinements and Experiments in Model Optimization

3.2.1 Baseline Model

- **Changes:** Used the starter notebook to implement a simple feedforward neural network.
- **Results:** Achieved a public score of 0.232 and ranked 621. This served as the baseline for further improvements.

3.2.2 Data Preprocessing and Feature Engineering

- **Objective:** Address issues with data scaling, categorical encoding, and class imbalance to enhance model interpretability and performance.
- **Key Changes:**
 - **MinMax Scaling:** Standardized numerical columns but showed minimal improvement.
 - **One-Hot Encoding :** Included categorical features, resulting in a notable jump in public score from 0.139 to 0.438 and an improvement in ranking to 576.
 - **Date Transformation :** Converted date columns into usable features but did not impact the score significantly.
 - **Oversampling:** Balanced the dataset using SMOTE, resulting in a slight score improvement but no ranking change.
 - **Feature Addition :** Added new features to better capture patterns in the data, improving the private score to 0.300.

3.2.3 Outlier Handling and Column Reduction

- **Changes:**
 - Handled outliers in numerical columns using the IQR method, reducing the noise caused by extreme values.
 - Dropped irrelevant ID columns to simplify the dataset.
- **Results:** Achieved a moderate increase in public and private scores, but the ranking remained consistent at 603.

3.2.4 Model Optimization and Regularization

Key Adjustments:

- **Early Stopping:** Tested different patience rates (5 and 10) to avoid overfitting.
- **Learning Rate Scheduling:** Introduced a scheduler with an initial rate of 0.001 but observed minimal impact.
- **Regularization Techniques:**
 - Added **Leaky ReLU** activation and **L2 Regularization**, improving the private score to 0.305.
 - Incorporated **Batch Normalization**, achieving a significant jump to 0.380.

3.2.5 Neural Network Architecture Tuning

- **Changes:**
 - Increased the number of neurons per layer to 128, capturing more complex patterns.
 - Experimented with different dropout rates (0.2, 0.3, 0.4) to reduce overfitting, with the best results observed at a dropout rate of 0.3.
 - Increased the number of epochs progressively (50, 100, 200), stabilizing model performance without overfitting.
- **Results:** Achieved the best private score of 0.466 with dropout and 50 epochs, ranking 631.

3.3 Result Visualization and Interpretation

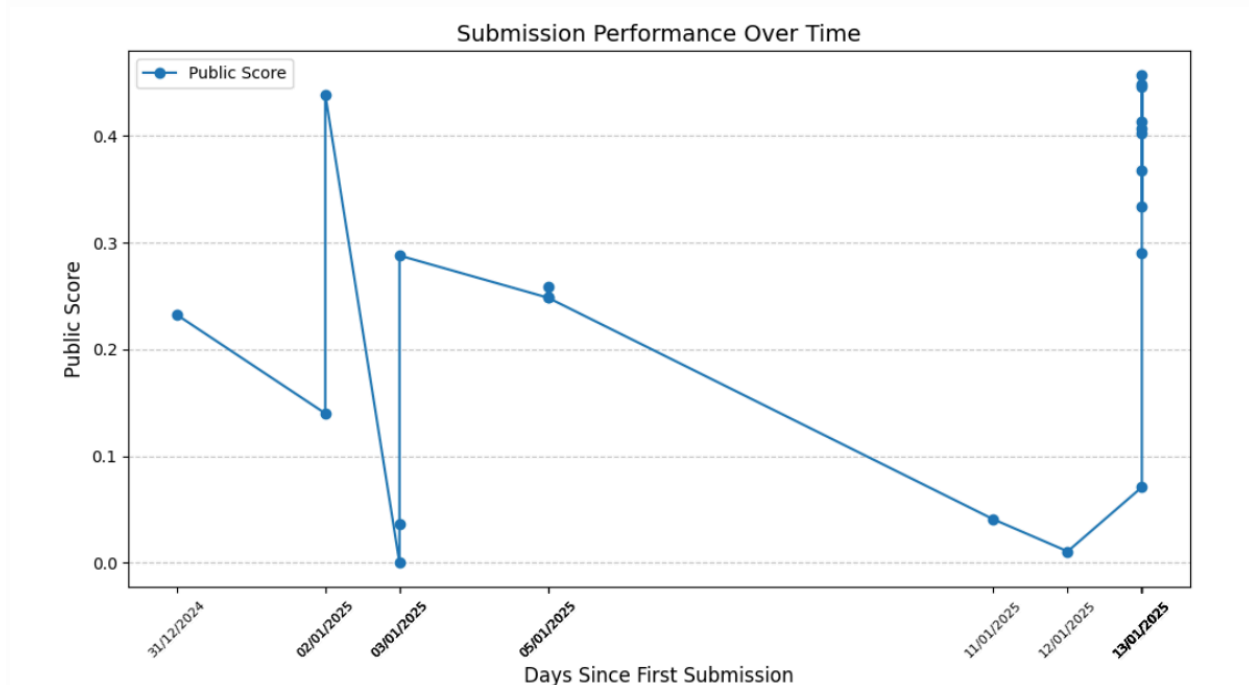


Figure 1. Line graph for Submission Performance over time.

Figure 1 shows a line graph to Visualize the submissions made over the period of 14 days against the public scores. Most submissions were made on 13/01/2025 and the highest recorded scores were achieved on 02/01/2025 and 13/01/2025.

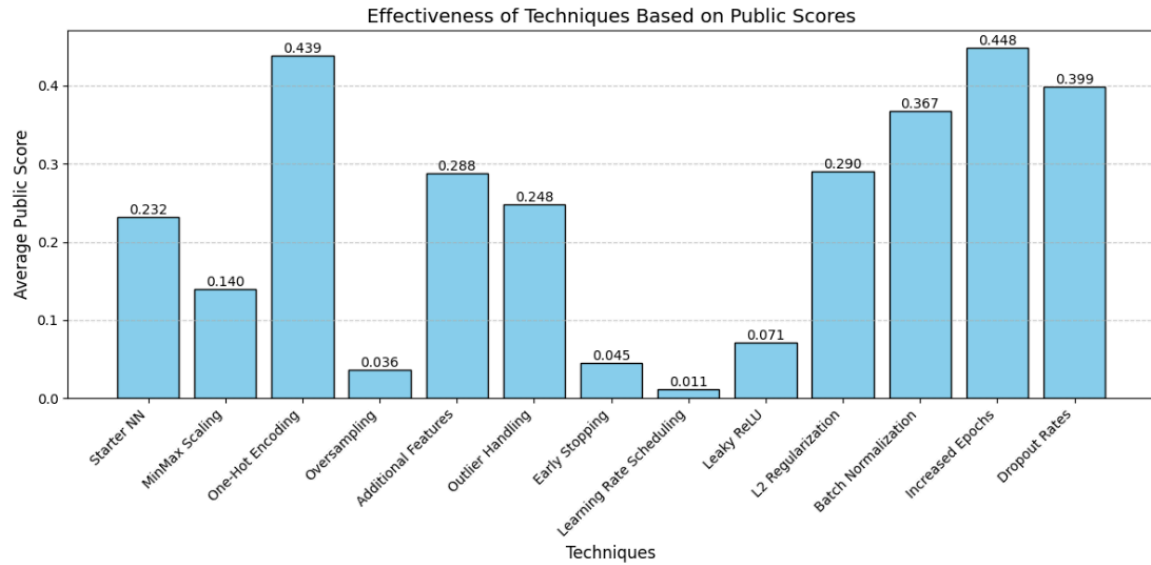


Figure 2. Bar graph showing Effectiveness of Techniques based on Public Scores.

From **Figure 2**, we notice the most impactful techniques were feature engineering (categorical encoding, feature addition), batch normalization, dropout and increasing the number of epochs. The least impactful technique was learning rate scheduling.

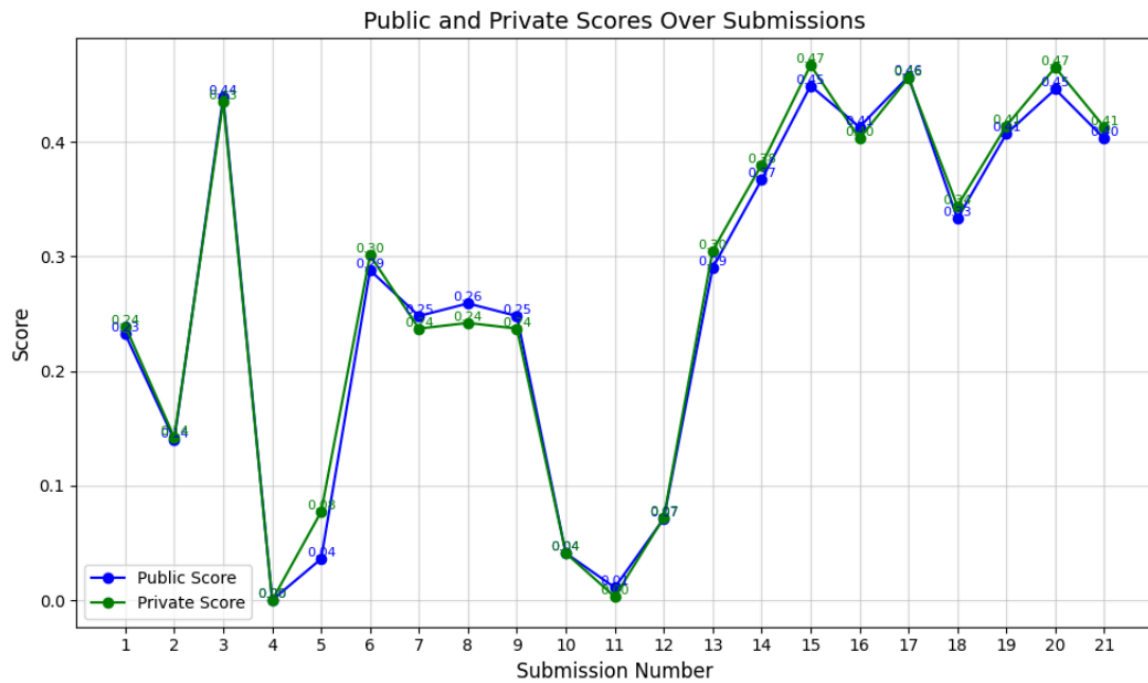


Figure 3. Line graph comparing private and public scores.

Figure 3 shows the gap between public and private scores is minimal across submissions, showing that the model generalizes similarly on both datasets.

3.4 Final Ranking

Our highest recorded ranking remained at 631, with a notable public score of 0.4487, achieved with increased epochs and later, a private score revealed to 0.466 as shown in **Figure 4**. The ranking on Zindi placed us in the middle to lower tiers, reflecting room for improvement in our approach.

<input type="checkbox"/>	YUBQZ32G	7 days ago	thinleyyc	submission_... ↓	0.448717948	0.466737064	—
--------------------------	----------	------------	-----------	----------------------------------	-------------	-------------	---

Figure 4. Highest recorded Submission.

However, the final submission that was chosen to be ranked was submission number 3 and we ranked 756 out of 901 submissions that were within the benchmark as shown in **Figure 5**.


RANK	USER	PUBLIC SCORE	PRIVATE SCORE	LAST SUBMISSION	# SUBMITTED
756	 kiwi Team	0.438709677	0.435058078	Go to placement 18 days ago	24
Benchmark		0.32752613240418120.3487064116985377			

Figure 5. Leaderboard Rank of Team Kiwi.

4.0 Conclusion / Future work

The project highlighted the importance of robust preprocessing techniques, such as oversampling, batch normalization, loss function, and dropout rates, which were essential for improving performance and stabilizing training. Effective feature engineering emerged as a critical factor for success. In addition to this, we failed to utilize the economic indicators dataset. Additionally, the project underscored the value of domain knowledge, demonstrating how incorporating relevant datasets and aligning machine learning techniques with problem-specific context can significantly enhance results.

- **Lessons Learned**

1. Feature Engineering: We did not utilize the economic indicators dataset, which could have significantly enriched the model's inputs. Incorporating domain-specific features might have yielded better performance.
2. Temporal Features: We lack experience with temporal data and did not explore temporal feature engineering. Leveraging time-dependent patterns could have been beneficial.
3. Model Fine-tuning: Many techniques, such as learning rate scheduling and L2 regularization, may have been under-optimized, limiting their effectiveness.

- **Shortcomings and Future Work**

Our simplistic feature engineering and limited experience with temporal features hindered the model's ability to capture complex patterns. Future efforts should focus on leveraging additional datasets like economic indicators, exploring temporal modeling techniques, and conducting deeper feature selection and engineering. With more time and resources, these improvements could significantly enhance results.