Predicting Bank Customer Churn with ANN Model

Sungwoo Noh   Jaejoong Kim

Listed alphabetically due to the equal contributions in proposal, coding, presentation, and final paper.

DS340 Introduction to Machine Learning and AI

Abstract

We present a deep learning model designed to predict customer churn using Artificial Neural Networks (ANNs) on bank customer data. The model employs a fully connected architecture with ReLU activation and dropout layers, trained on a modeling dataset containing customer demographics, account features, and activity metrics. Our preprocessing pipeline incorporates feature encoding, normalization, and SMOTE to handle class imbalance. Compared to traditional methods: Logistic Regression and Random Forest, the ANN model achieves superior accuracy and recall, demonstrating its ability to detect complex patterns in customer behavior. We validate the model using K-fold cross-validation and highlight its generalizability through extensive evaluation metrics.

1 Introduction

Customer churn poses a significant challenge for businesses, particularly in the financial sector, where retaining clients directly impacts profitability. According to Bain & Company[1], reducing churn by just 5% can lead to a 25–95% increase in operating profit. This project aims to predict customer churn using Artificial Neural Networks (ANNs) applied to banking customers dataset[2]. By identifying customers likely to leave, businesses can devise targeted strategies to improve retention and minimize revenue losses.

Our analysis began with exploratory data analysis (EDA) to uncover trends and correlations. Key insights were derived from visualizations, including a correlation heatmap that showed weak multicollinearity among features, indicating they provide unique predictive power for churn modeling (Figure 1). Categorical variable trends with **s**tacked bar charts revealed higher churn rates among German customers, inactive members, and those with limited tenure or fewer products. Boxplots for numerical features highlighted that older customers and those with significant balances are more likely to churn, while estimated salary had little impact. Customer Distribution demonstrated that 20.4% of customers in the dataset had churned, emphasizing the class imbalance challenge (Figure 2).

[1] Bain & Company. "Retaining Customers Is the Real Challenge." in & Company Insights.

[2] Kaggle. "Churn Modelling Data." Kaggle Datasets,https://www.kaggle.com/datasets/shubh0799/churn-modelling/data

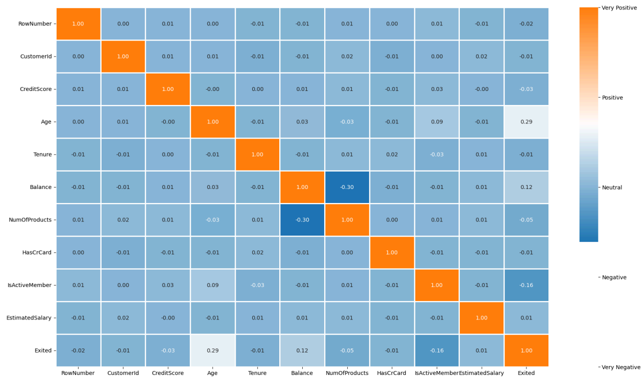


Figure 1: Correlation Heatmap

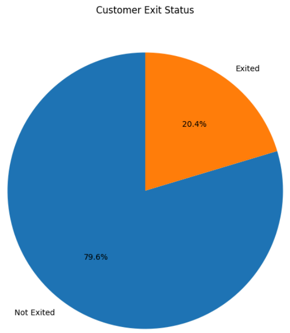


Figure 2: Distribution of Customer Churn

Our methodology involves rigorous model evaluation and comparison to establish the effectiveness of the ANN. The primary focus is on optimizing the ANN architecture through hyperparameter tuning, feature selection, and handling class imbalances using techniques like SMOTE (Figure 3)[3]. Evaluation metrics such as accuracy, precision, recall, and F1-score were extensively analyzed to validate model performance.

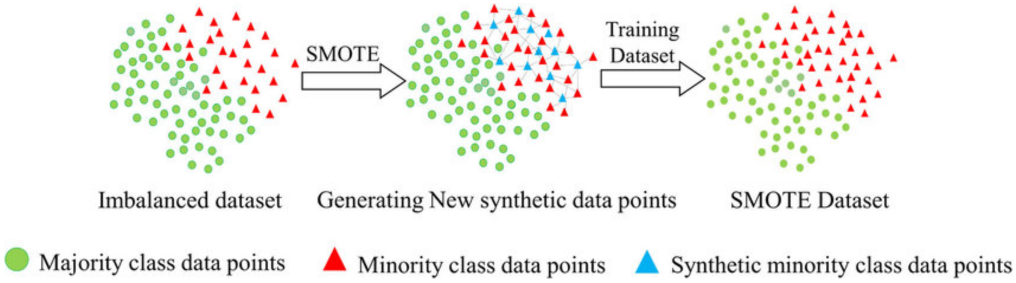

SMOTE
Training Dataset
Imbalanced dataset
Generating New synthetic data points
SMOTE Dataset
Majority class data points
Minority class data points
Synthetic minority class data points

Figure 3: Overview of SMOTE Technique

We benchmarked the ANN against traditional models, including Logistic Regression and Random Forest, demonstrating its superior performance in identifying churn patterns. By leveraging cross-validation and advanced evaluation techniques, we ensured the robustness and generalizability of the ANN model (Figure 4)[4]. This work provides actionable insights for mitigating churn and highlights the value of deep learning in real-world financial applications.

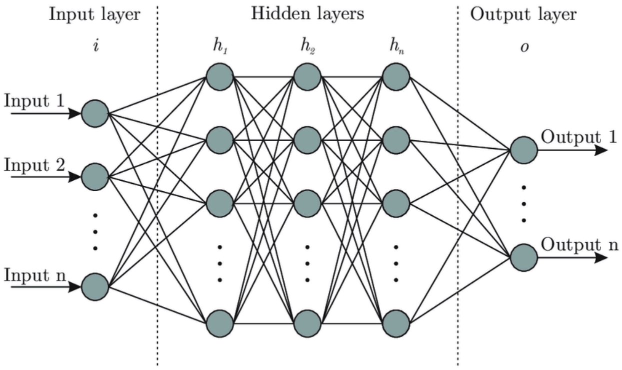


Figure 4: Overview of Artificial Neural Network

[3] AIMind. "Synthetic Minority Over-Sampling Technique: Empowering AI through Imbalanced Data Handling." AIMind Publications.

[4] ResearchGate. "Artificial Neural Network Architecture (ANN)." ResearchGate.

2 Methodology

To develop a predictive model for customer churn, we employed a systematic approach involving data preprocessing, feature selection, model architecture design, and evaluation. The dataset, consisting of 10,000 samples with 14 features, was preprocessed by encoding categorical variables: ‘Geography’ and ‘Gender’, normalizing numerical features: ‘CreditScore’ and ‘Balance’, and addressing class imbalance using SMOTE.

Feature selection was conducted using Lasso Regularization and Forward Selection, narrowing the input features to ‘CreditScore’, ‘Gender’, ‘Age’, ‘Balance’, and ‘IsActiveMember’ for optimized performance. The data was split into training and testing sets, with the training set further scaled using StandardScaler to enhance model convergence.

The ANN architecture was designed with one input layer, two hidden layers, and one output layer. Each hidden layer employed ReLU activation, while the output layer used a Sigmoid activation function for binary classification. To prevent overfitting, Dropout layers were incorporated, and Batch Normalization was applied to stabilize learning (Figure 5). Hyperparameter tuning was performed using GridSearchCV with K-fold cross-validation to identify the best combination of optimizers (Adam, RMSprop), activation functions (ReLU, Tanh)[5], dropout rates, and batch sizes. The final model was trained with the optimal parameters: 32 neurons per hidden layer, a dropout rate of 0.1, the RMSprop optimizer, and a batch size of 32.

Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. Cross-validation and a confusion matrix analysis ensured the robustness and reliability of the results. These steps culminated in a high-performing ANN model tailored for predicting customer churn.

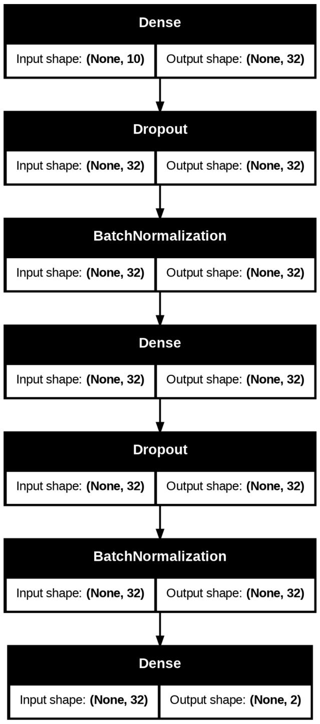


Figure 5: ANN Architecture

3 Results

The evaluation of the ANN model demonstrated its strong performance in predicting customer churn, supported by both numerical metrics and visualizations. The final model achieved a validation accuracy of 86.76%, outperforming baseline models like Logistic Regression and Random Forest. Through rigorous hyperparameter tuning using GridSearchCV, the optimal configuration of the model was determined: an RMSprop optimizer, ReLU activation function, a dropout rate of 0.1, and a batch size of 32.

Further metrics reinforced the model's reliability, achieving a precision of 77.22%, a recall of 40.65%, and an F1-score of 53.26% on the test dataset. These results highlight the model's ability to detect churned customers while maintaining a reasonable balance between precision and recall. The application of SMOTE during preprocessing played a significant role in improving the detection of the minority churn class, as confirmed by the confusion matrix analysis.

[5] Oppermann, Artem. "Activation Functions in Deep Learning: Sigmoid, Tanh, ReLU." Artem Oppermann Blog.

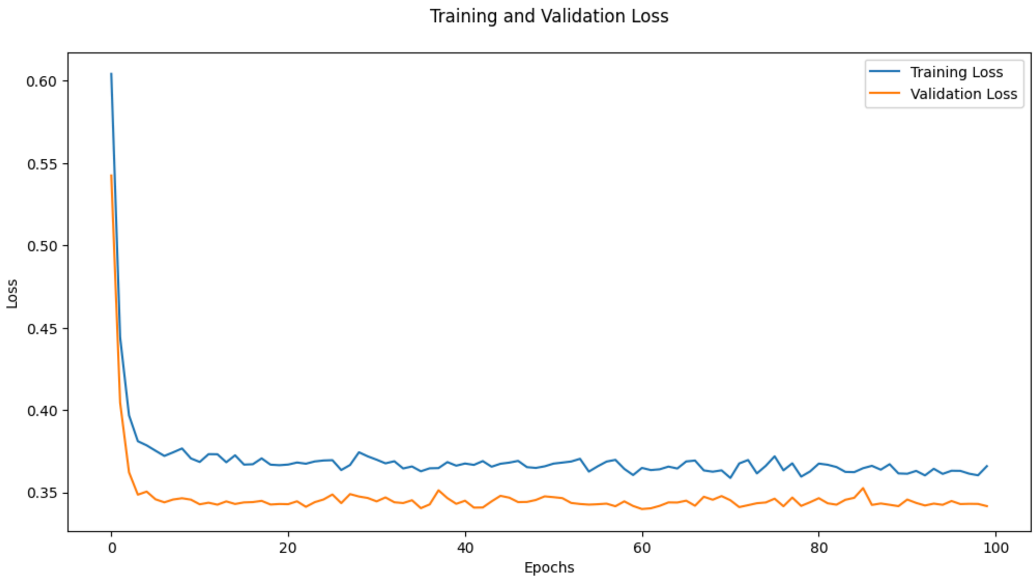


Figure 6: Training and Validation Loss During Model Training

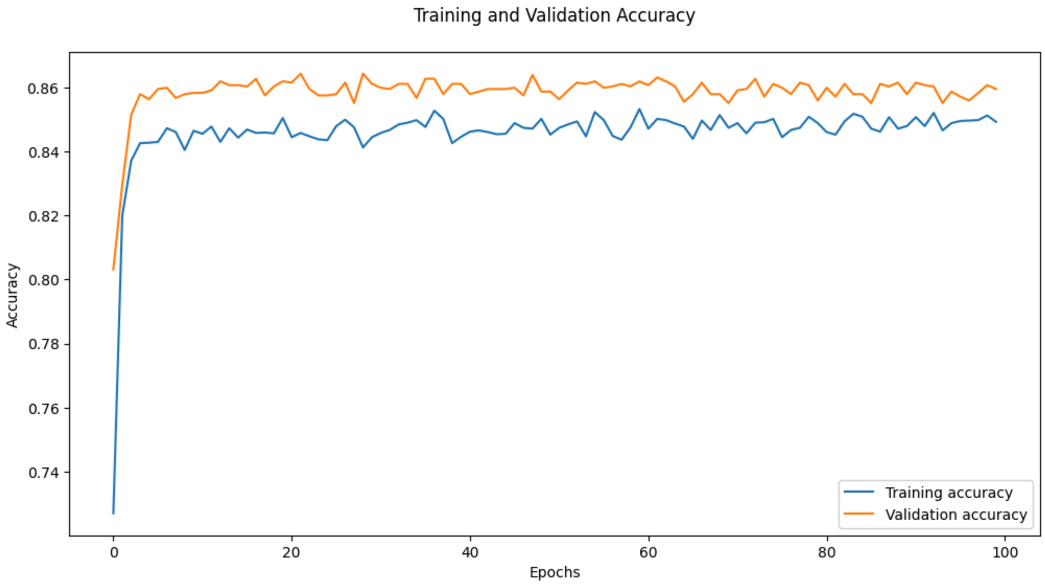


Figure 7: Training and Validation Accuracy During Model Training

Visualizations of the training process provide further insights into the model's performance. As shown in Figure 6, the training and validation loss steadily decreased in the early epochs, converging to a stable point without signs of overfitting. This indicates that the model effectively minimized the binary cross-entropy loss. Figure 7 illustrates the training and validation accuracy curves, with the validation accuracy stabilizing around 86.76%, slightly exceeding the training accuracy. This demonstrates the model's ability to generalize well to unseen data.

These results, supported by both quantitative metrics and qualitative insights from the training curves, establish the ANN as a robust and reliable solution for predicting customer churn. Its capacity to capture complex, non-linear relationships in the data underscores its superiority over traditional models.

4 Experiments

This chapter goes into detail about experiments to better understand how our experiments function and analyze the results for further improvements. We conducted four different experiments to improve our initial model and achieve our primary project objective: predicting customer churn rates and identifying the key features influencing churn. Our approach included feature selection, data balancing, hyperparameter tuning, and hypothesis testing. These steps were designed to enhance the model's performance and generate meaningful insights.

4.1 Feature selection

We employed both Lasso regularization and forward selection to identify the most important features. We then ran separate models using the features selected by each method and compared their performance to our baseline model, using accuracy as the evaluation metric (Figure 8).

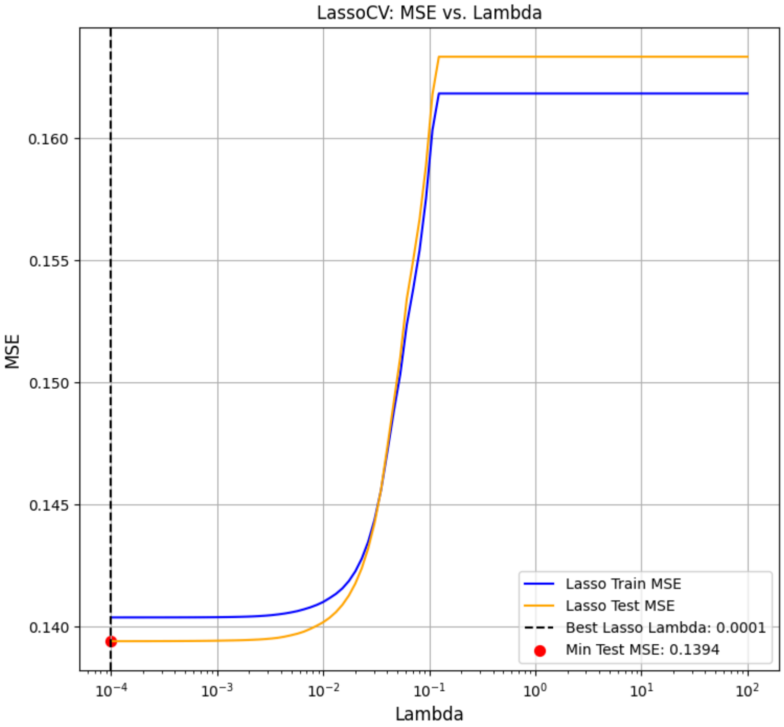


Figure 8: Relationship Between Lambda and MSE in Lasso Regression (LassoCV)

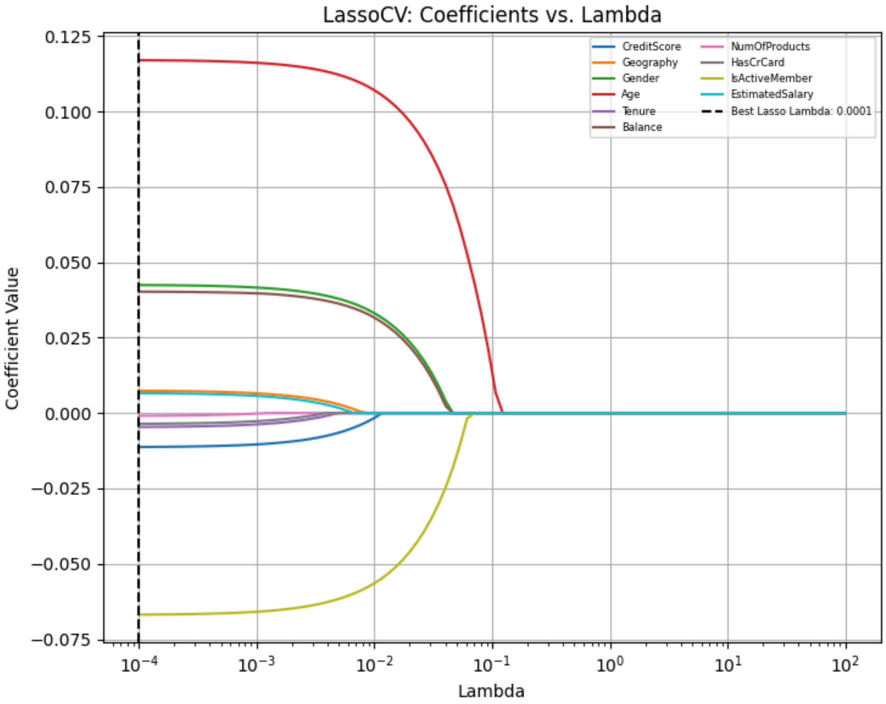


Figure 9: Impact of Lambda on Feature Coefficients in Lasso Regression (LassoCV)

For Lasso regularization, we utilized LassoCV with 5-fold cross-validation to determine the optimal lambda that minimizes the mean squared error (MSE). The best lambda was found to be approximately 10^-4. Based on the graph (Figure 9), the features selected by Lasso Regularization are 'CreditScore,' 'Gender,' 'Age,' 'Balance,' and 'IsActiveMember,' with 'Age' having the largest coefficient. The model's accuracy using these selected features was 0.8340. Additionally, the datasets were pre-scaled to ensure compatibility with Neural Networks, as scaling is a necessary preprocessing step for their use.

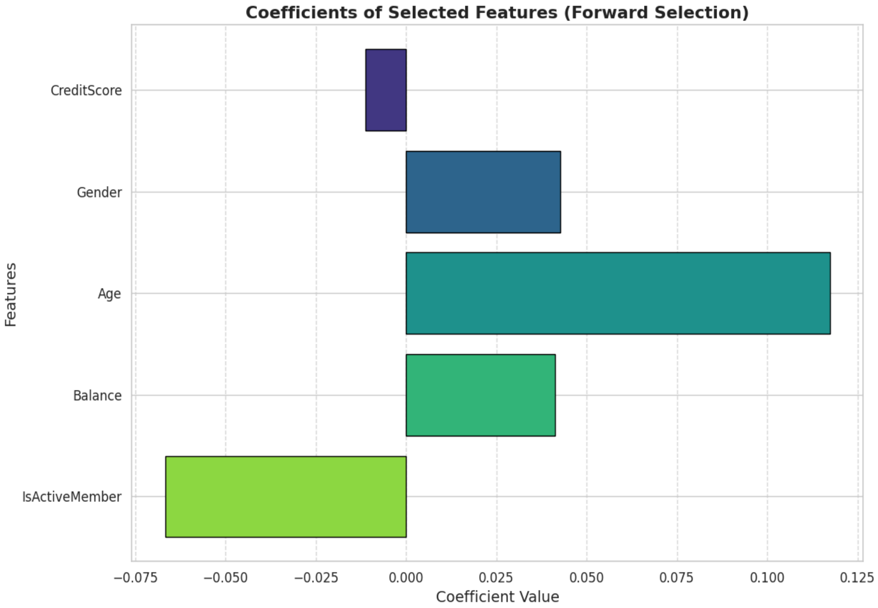


Figure 10: Coefficient of Selected Features (Forward Selection)

For forward selection, we employed a SequentialFeatureSelector with a linear regression model, using 5-fold cross-validation and R2 as the scoring metric. The top five features selected by forward selection are 'Geography,' 'Gender,' 'Age,' 'Balance,' and 'IsActiveMember' (Figure 10). The validation accuracy of the model with these features was 0.8392, compared to the baseline model's accuracy of 0.8672. The top 5 features selected by forward selection are 'Geography', 'Gender', 'Age', 'Balance', and 'IsActiveMember', and their validation accuracy is 0.8344, whereas our baseline model, random forest, has an accuracy of 0.867. This is surprising as we expected the model with selected features to show higher accuracy than the baseline model. This could indicate that the ANN model needs to be fed with more features as the model can not capture patterns of the data with fewer features. Instead of overfitting, our model may have been underfitting due to the reduced feature set, leading to its inability to generalize the data effectively. This suggests that including more features could enhance the model's performance with a richer representation of the data.

4.2 Balance imbalanced Data

The dataset showed an imbalance in the target variable, with 20% of observations representing one class and 80% representing the other. This uneven distribution could result in biased and misleading outcomes. To address this issue, we applied the SMOTE (Synthetic Minority Oversampling Technique) method to resample the target variable, ensuring that the minority class was balanced to match the size of the majority class. After applying SMOTE, the classes had equal observations, allowing for a more balanced analysis without potential biases from the original imbalance. Examining the graphs, we can observe the stark imbalance in the target variable before applying SMOTE and the equalized distribution afterward.

The validation accuracy of the model trained on a balanced dataset is 0.8274, which is unexpectedly lower than the baseline model's accuracy of 0.8672. The imbalance in target variables often leads to biased outcomes and low model performance. However, in our model, the baseline model ran with an imbalanced dataset shows better performance. This suggests that the imbalance in the target variable might represent a unique pattern inherent to this dataset. Instead of being a limitation, this imbalance appears to provide meaningful insight into the data, indicating that the original distribution may better reflect the real-world dynamics of the problem.

While SMOTE provides a means to address potential biases, this case highlights the importance of considering whether the original distribution carries valuable information about the underlying data characteristics

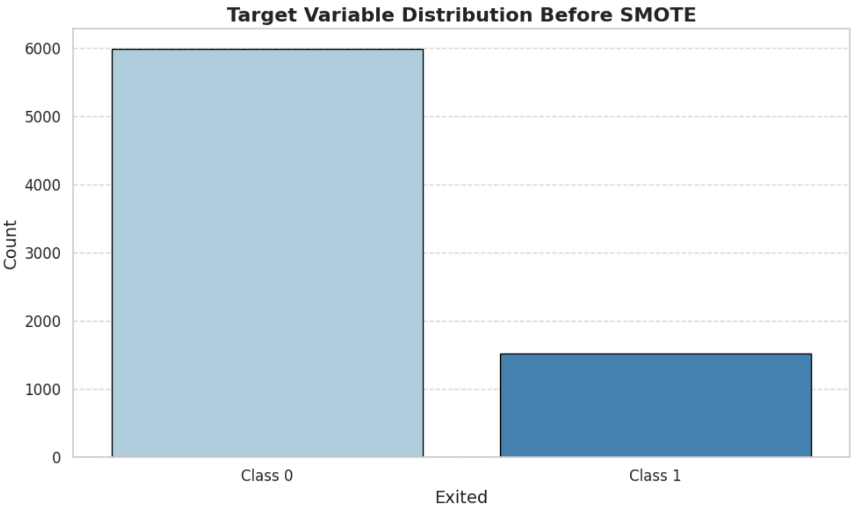


Figure 11: Distribution of Target Variable before SMOTE

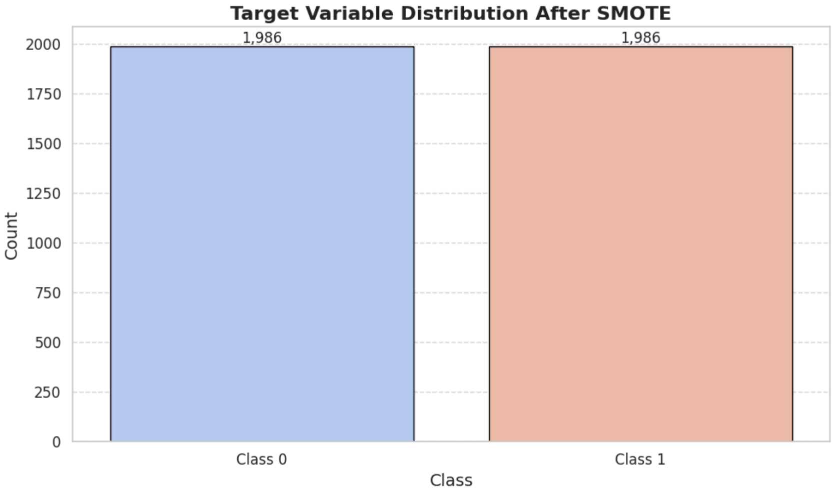
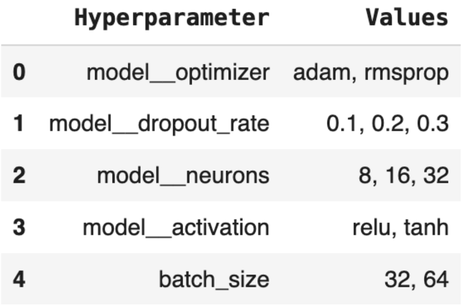


Figure 12: Target Variable Distribution Before / After SMOTE

4.3 Hyperparameter Tuning

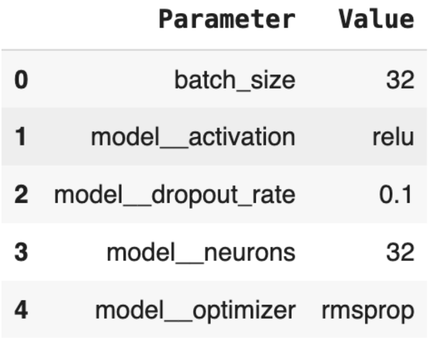
Neural networks are highly sensitive models with numerous hyperparameters that significantly influence their performance. To optimize the model and maximize its performance, we utilized GridSearchCV. This method allowed us to define a grid of hyperparameters and evaluate the model's performance by iterating through each combination of parameters across validation sets. The parameter grid included the optimizer, dropout rate, number of neurons, activation function, and batch size, as detailed in the figure below. The grid search was conducted with a 5-fold CV to ensure robust evaluation and selection of the best hyperparameter combination.



	Hyperparameter	Values
0	model__optimizer	adam, rmsprop
1	model__dropout_rate	0.1, 0.2, 0.3
2	model__neurons	8, 16, 32
3	model__activation	relu, tanh
4	batch_size	32, 64

Figure 13: Table of Parameter Grid

The process helped us identify the best set of hyperparameters based on accuracy. The best parameters identified through GridSearchCV included: 'batch_size': 32, 'model__activation': 'relu', 'model__dropout_rate': 0.1, 'model__neurons': 32, 'model__optimizer': 'rmsprop'. The model trained with these parameters achieved a validation accuracy of 0.8708, slightly higher than the baseline model's accuracy of 0.8672. While this represents a modest improvement, the difference is so small that it does not conclusively demonstrate that the optimized model outperformed the baseline model. This could indicate that the baseline model was already well-optimized, leaving limited room for improvement. Alternatively, it might suggest that other factors, such as feature engineering or a deeper exploration of the model architecture, could have a more significant impact on performance than hyperparameter tuning alone. However, taking the fact that balancing the dataset and feature selection did not improve the model well, other factors might have limited impact as well. Additionally, the small improvement could reflect the inherent limitations of the dataset or the difficulty of the problem itself. Nonetheless, there remains potential for further improvement through the remaining experiments or deeper exploration of the model's architecture and training strategies.



	Parameter	Value
0	batch_size	32
1	model__activation	relu
2	model__dropout_rate	0.1
3	model__neurons	32
4	model__optimizer	rmsprop

Figure 14: Table of Best Parameters

4.4 Hypothesis Testing

Hypothesis testing was employed to identify the most important features, aligning with our project goal of understanding the factors influencing customer churn and reducing churn rates. Specifically, we focused on the has_credit variable since intuitively customers who have credit care tend not to exit the bank. We built two models: one including the has_credit variable and another excluding it. By comparing the accuracy of these models, we evaluated the importance of the feature. Additionally, we utilized SHAP (SHapley

Additive exPlanations) to assess the variable's feature importance and performed a hypothesis test using linear regression to obtain the p-value, further confirming its significance.

According to the hypothesis test we conducted, we failed to reject the null hypothesis that there is no significant difference between the models with and without the has_credit_card feature. The p-value of 0.993 is extremely high, indicating strong statistical insignificance. Additionally, the distribution of SHAP values for has_credit_card shows that most values are centered around 0.000, suggesting that this feature has minimal importance in the model. These findings, supported by SHAP analysis, make it evident that has_credit_card has no meaningful impact on predicting the churn rate. Moreover, the accuracy of the model without the has_credit_card feature is 0.8692, which is very similar to the best model's accuracy. This further supports the conclusion that including has_credit_card does not improve the model's performance.

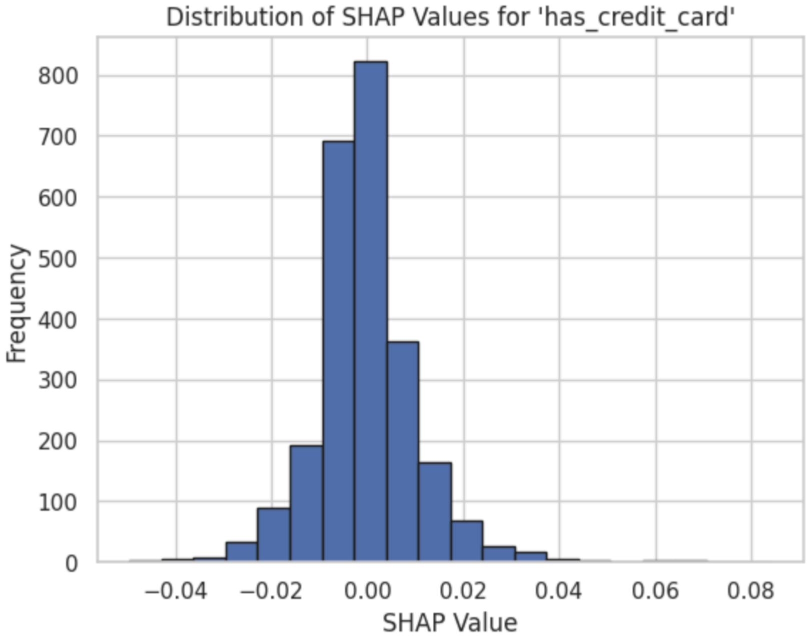
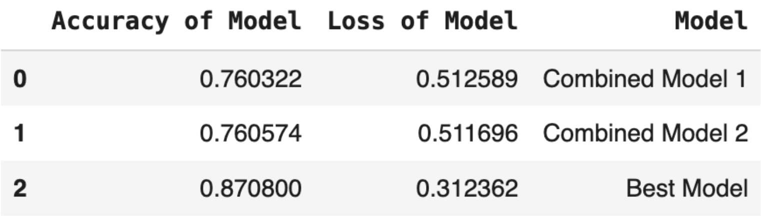


Figure 15: Distribution of SHAP Values for feature

4.5 Combined Experiments

The purpose of the combined experiments is to enhance model performance, as earlier experiments produced unexpectedly poorer results despite being designed to improve accuracy. These experiments aim to identify the factors contributing to the decline in performance. We have designed two models for comparison. The first is a model using features selected by Lasso, with the best hyperparameters identified by GridSearchCV, trained on a balanced dataset. The other is a model using all features, with the best hyperparameters, trained on a balanced dataset. These models will be compared to the model with the best parameters, which includes all features and is trained on an imbalanced dataset. The features for the first model were selected using Lasso regularization. Since we can assume that the models selected by Lasso and forward selection have negligible differences, the specific selection method should not significantly impact the results. This comparative analysis will help us gain a deeper understanding of the feature selection process and its effect on model performance. This comparative analysis will help us better understand the impact of feature selection data balancing, and hyperparameter optimization on model performance.



	Accuracy of Model	Loss of Model	Model
0	0.760322	0.512589	Combined Model 1
1	0.760574	0.511696	Combined Model 2
2	0.870800	0.312362	Best Model

Figure 16: Comparison Table of Combined Model 1,2 and Best Model

The results show that the best model achieves the highest accuracy, while Combined Model 1 and Combined Model 2 demonstrate similar performance, both with an accuracy of approximately 76%. The key difference between these two combined models is that Combined Model 1 uses selected features, while Combined Model 2 utilizes all features. Both models share the same conditions of a balanced dataset and optimized parameters.

In contrast, the best model uses all features, the best parameters, and an imbalanced dataset. These findings suggest that the imbalance in the dataset captures a meaningful pattern inherent to this data. Rather than being a limitation, the imbalance appears to reflect an underlying structure in the data that contributes to the superior performance of the best model.

5 Conclusion

From our results, we conclude that Artificial Neural Networks (ANNs) demonstrate substantial potential for predicting customer churn in the banking sector. Our analysis identifies key factors such as age, balance, and geography as the most influential predictors of customer churn, consistent with the insights gained from exploratory data analysis (EDA). Specifically, older customers, those with higher account balances, and German customers showed higher churn rates, providing actionable insights for targeted retention strategies.

5.1 Performance of ANN vs. Traditional Models

The ANN model consistently outperformed traditional models like Logistic Regression and Random Forest in terms of accuracy and recall, showcasing its ability to capture non-linear relationships and complex patterns in the data.

5.2 Impact of Dataset Imbalance

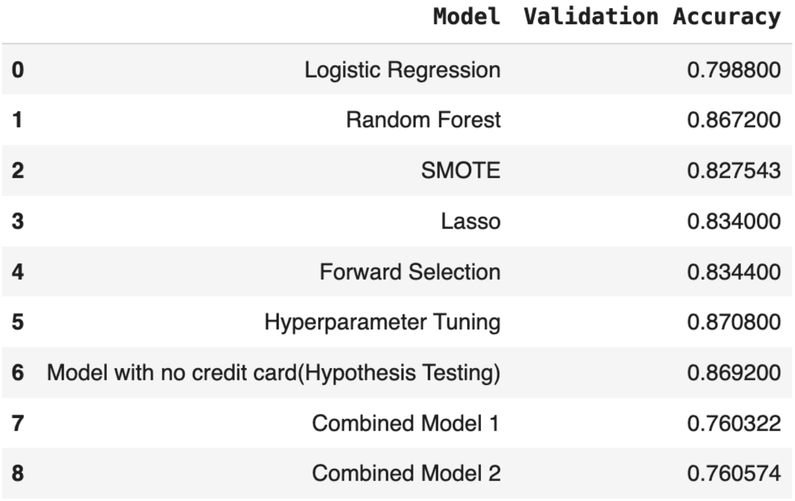
Surprisingly, the best-performing model used an imbalanced dataset, suggesting that the imbalance may reflect meaningful patterns inherent in customer behavior. Efforts to balance the dataset with SMOTE, while theoretically sound, led to lower accuracy, indicating that the original class distribution might better represent real-world dynamics.

5.3 Feature Selection and Hyperparameter Tuning

While feature selection methods like Lasso and forward selection identified critical predictors, reducing the feature set negatively impacted performance. This suggests that ANNs benefit from richer feature representations. Hyperparameter tuning provided only marginal improvements, indicating that the baseline ANN model was already near optimal.

5.4 Model Robustness and Generalization

The use of cross-validation and rigorous evaluation metrics confirmed the model's reliability. However, additional experiments integrating selected features and balanced datasets did not surpass the baseline model, underscoring the complexity of the problem and potential limitations in the dataset



	Model	Validation Accuracy
0	Logistic Regression	0.798800
1	Random Forest	0.867200
2	SMOTE	0.827543
3	Lasso	0.834000
4	Forward Selection	0.834400
5	Hyperparameter Tuning	0.870800
6	Model with no credit card(Hypothesis Testing)	0.869200
7	Combined Model 1	0.760322
8	Combined Model 2	0.760574

Figure 17: Comparison Table for Accuracy of All Models

5.5 Lessons Learned

Our findings emphasize the importance of considering the implications of dataset imbalance. In some cases, preserving the original distribution may better capture real-world behavior than forcing balance through techniques like SMOTE. While our experiments highlighted key predictive features, further exploration of feature engineering such as creating interaction terms or leveraging external data may enhance the model's capacity to generalize and capture nuanced patterns. Despite the ANN model's superior performance, the modest recall score (40.65%) indicates room for improvement in identifying churned customers. This highlights the need for alternative architectures or advanced techniques, such as ensemble methods or transfer learning. In summary, while our ANN model demonstrates robust performance and valuable insights, the challenges encountered point to opportunities for deeper data analysis, feature enhancement, and exploration of advanced modeling approaches. However, in terms of model performance and predictions, none of our experiments outperformed the baseline model. This suggests the need for further and deeper analysis of the model and dataset to uncover additional insights and improve performance.

References

[1] AIMind. "Synthetic Minority Over-Sampling Technique (SMOTE): Empowering AI through Imbalanced Data Handling." *AIMind Publications*.

[2] Bain & Company. "Retaining Customers Is the Real Challenge." *Bain & Company Insights*, https://www.bain.com/insights/retaining-customers-is-the-real-challenge/.

[3] Oppermann, Artem. "Activation Functions in Deep Learning: Sigmoid, Tanh, ReLU." *Artem Oppermann Blog*.

[4] ResearchGate. "Artificial Neural Network Architecture (ANN)." *ResearchGate*.

[5] Kaggle. "Churn Modelling Data." Kaggle Datasets, http://www.kaggle.com/database/shubh0799/churn-modelling/data.

[6] OpenAI. "ChatGPT for grammar correction and Code Optimization."

[7] Google. "Gemini for code snippet and assistance."