Bank Churn Prediction

Introduction to Neural Networks
AI-ML

Oct 14th, 2023

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

Geographical Focus: Germany

- Tailored Marketing Campaigns: Given that Germany shows a higher churn rate, targeted marketing campaigns should be launched to improve customer satisfaction and reduce churn.
- Competitive Analysis: Conduct a comprehensive market study to understand what competitors in Germany are offering, and align your services and features accordingly.

Gender-Specific Strategies: Focus on Women

- Women-Centric Services: Launch services and products that are tailored to women's financial needs to improve retention among this demographic.
- Gender-Based Marketing: Use marketing campaigns that resonate with women to make them feel valued and understood by your institution.

Executive Summary

Activity Level

- Engagement Programs: Implement programs to encourage more frequent use of banking services, not just credit cards. Higher activity levels could lead to lower churn rates.
- Loyalty Rewards: Introduce a points-based rewards system for active users. Customers accumulating points could redeem them for various perks, encouraging continued usage.

Age-Specific Programs

- Retirement Planning Services: Given that older individuals are more likely to churn, offering specialized retirement planning services could be beneficial.
- Tourist-Focused Services: Offer specialized financial products for older tourists in Germany, Spain, and France, possibly in partnership with travel agencies.
- Age Segmentation: Focus marketing efforts on age groups showing higher churn rates, such as those between 40 and 50 years old.

Executive Summary

Balance-Based Programs

- Premium Services for High-Balance Accounts: For customers with higher balances, consider introducing premium services or account types that offer more personalized services or higher interest rates.
- Exclusive Offers: High-balance customers could be offered exclusive deals, such as lower fees for overseas transactions or higher interest rates on fixed deposits, to encourage them to maintain their account balance.

By implementing these strategies, the bank can target the most crucial factors contributing to customer churn. A multi-faceted approach that addresses geographical, gender, age, and balance factors can provide a robust framework for significantly improving customer retention for those are the variables that has some correlation found in the EDA stage of the analysis.

Business Problem Overview and Solution Approach

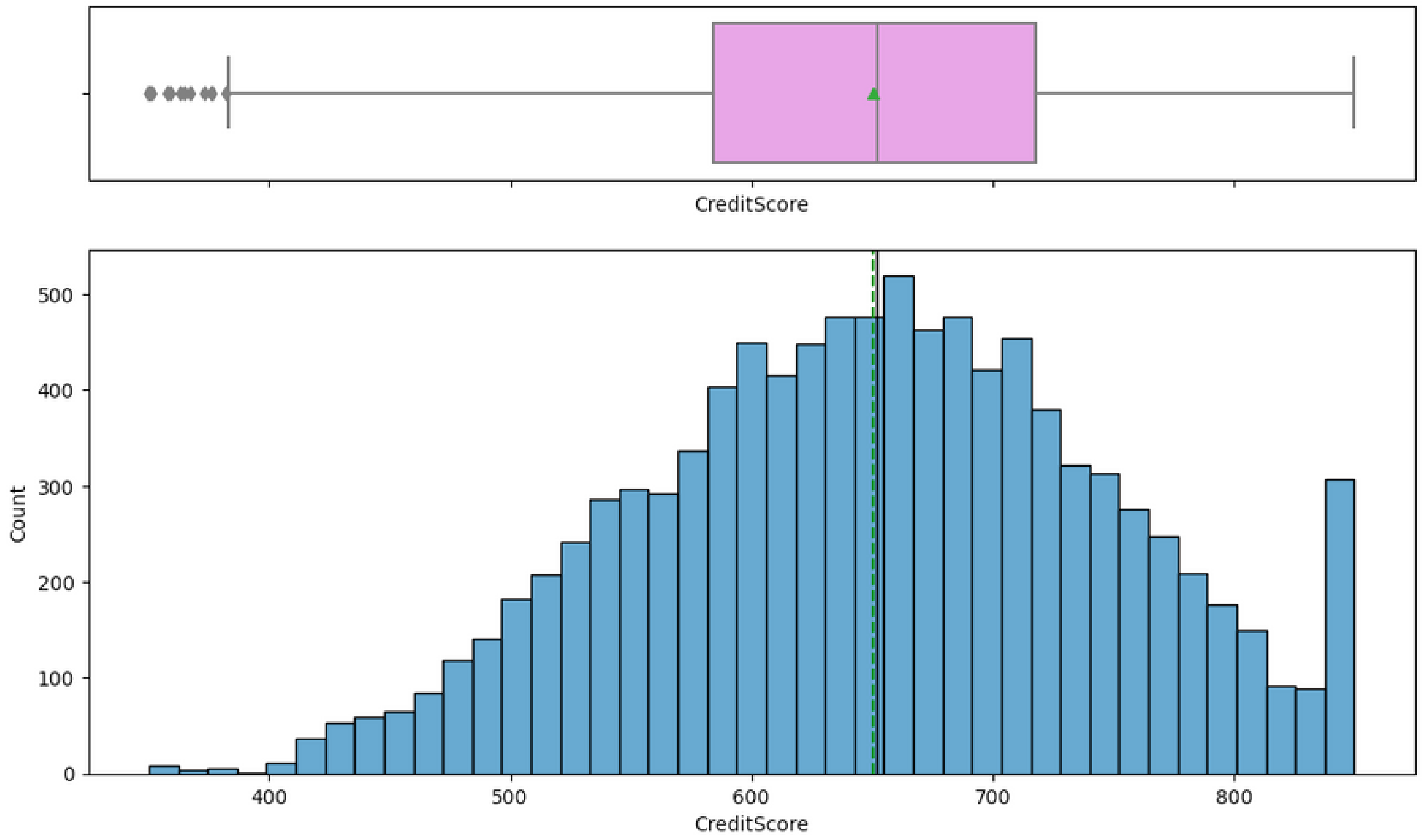
In service-oriented industries like banking, customer churn—defined as customers discontinuing their current service to switch to another provider—poses a significant challenge. Understanding the factors that influence a customer's decision to leave is critical. This knowledge allows management to prioritize and target their efforts effectively, aiming to enhance those service aspects that are most impactful in retaining customers.

In order to tackle the challenge of customer churn in the banking industry, we employed a data-driven approach. Our dataset comprises various features that could potentially influence a customer's decision to churn, such as account balance, credit score, and tenure with the bank. We began by conducting an exploratory data analysis to gain insights into the data and identify patterns. Following this, we preprocessed the data by dropping some features, scaling numerical features, and encoding categorical variables.

Business Problem Overview and Solution Approach

The core of our approach was to develop a predictive model using neural networks. We constructed several architectures and optimized them through hyperparameter tuning. Specifically, we utilized techniques such as dropout for regularization and early stopping to prevent overfitting. To improve the model's performance in predicting the minority class, we applied the Synthetic Minority Over-sampling Technique (SMOTE).
We evaluated each model using metrics like accuracy, precision, recall, and F1-score, placing special emphasis on recall due to its business importance. The models were also validated using ROC-AUC curves to find the optimal threshold for classification. The best-performing model was then selected based on these evaluations.
This methodology not only predicts potential churn but also provides insights into how different features impact customer retention, thus allowing management to make informed decisions.

EDA Results: Credit Score

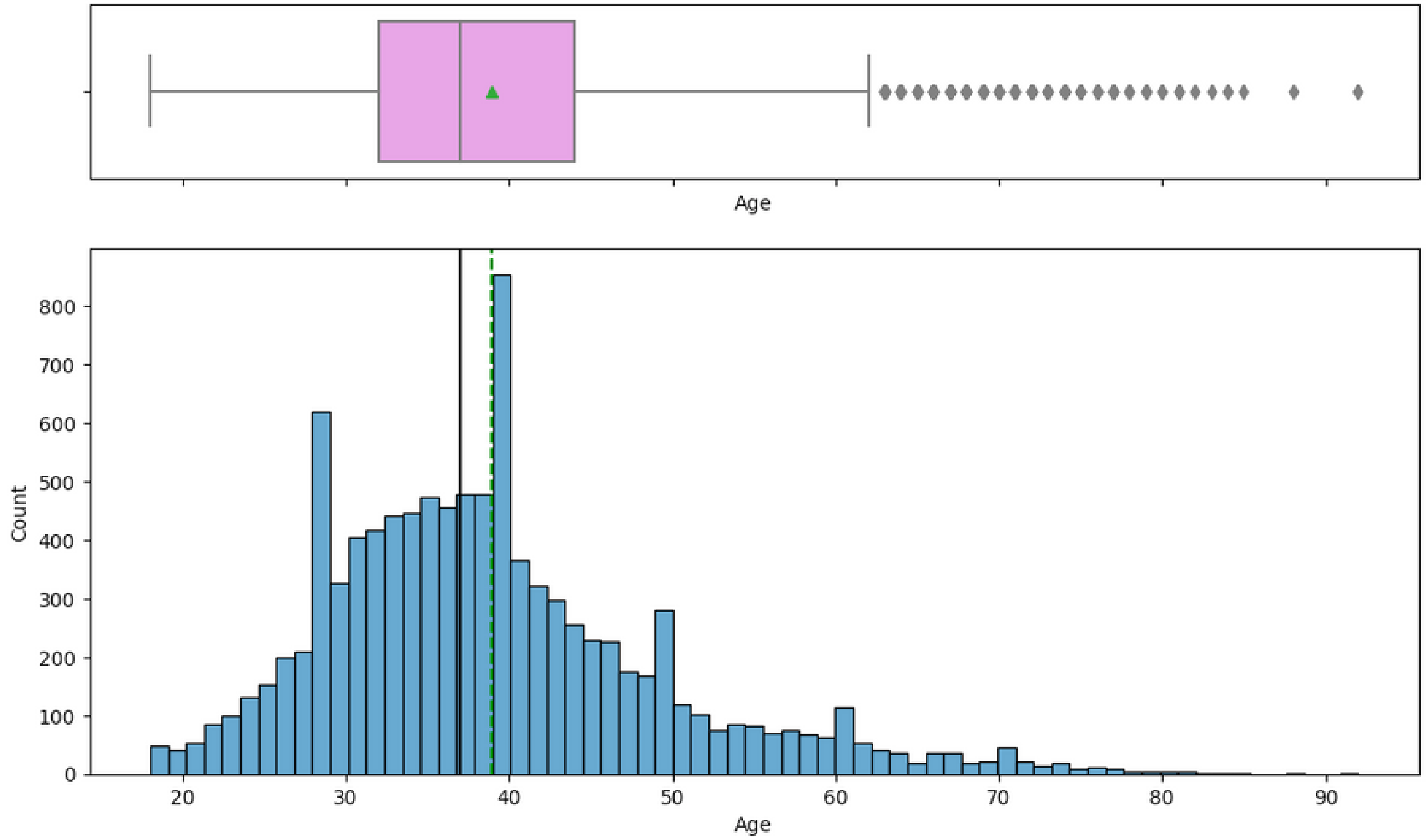


The average credit score among our customers is $650.53, with a standard deviation of $96.65, indicating a fairly broad range of creditworthiness. The lowest credit score recorded is $350, which is significantly below the average and we can observe outliers in the low side of the values.

EDA Results: Credit Score

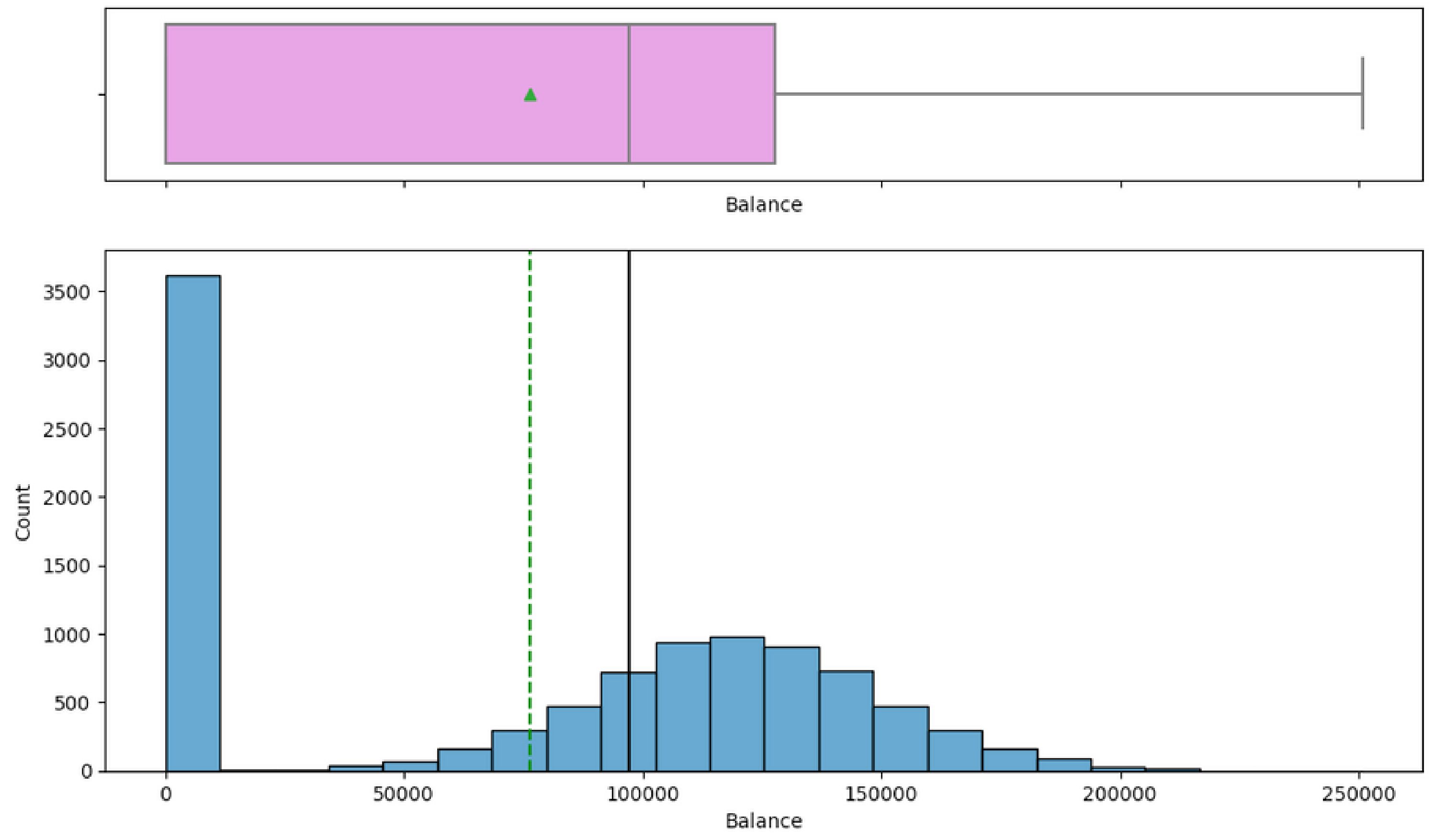
The first quartile falls at a score of $584, the median is at $652, and the third quartile is at $718, showing that most customers have credit scores clustered around the average. The maximum credit score observed in our dataset is the optimal $850. This metric enriches our understanding of the financial health of our customer base and can be particularly insightful for risk assessments and customer segmentation.

EDA Results: Age



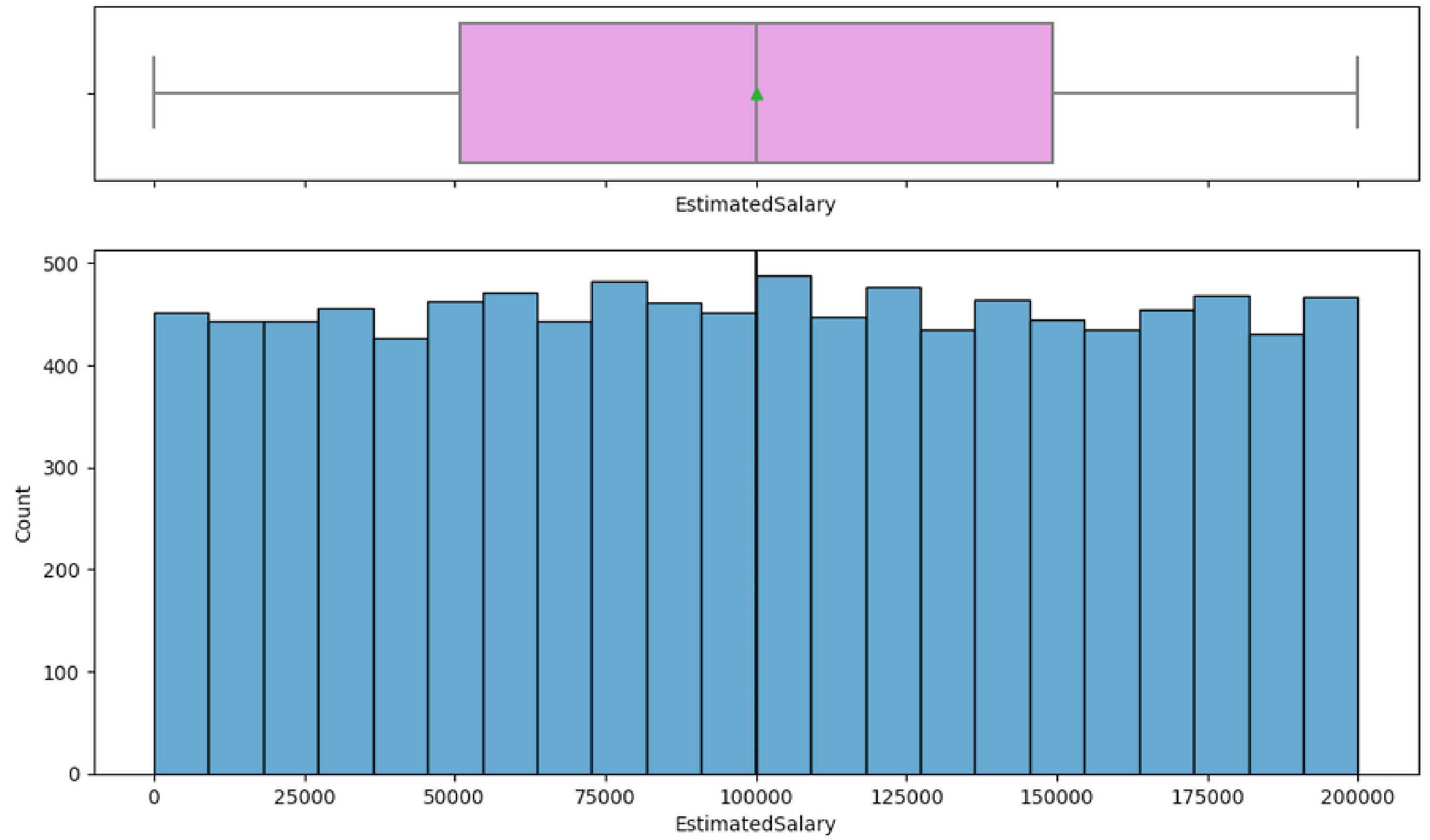
Our customer base has an average age of 38.92 years, with a standard deviation of 10.48, indicating a moderate variation in age. All clients are adults, with the youngest being 18 years old. The age distribution is skewed towards younger individuals; 25% are 32 years or younger, 50% are up to 37 years, and 75% are not older than 44 years. The eldest customer is 92 years old.
Customers above around 60 years old are considered outliers.

EDA Results: Balance



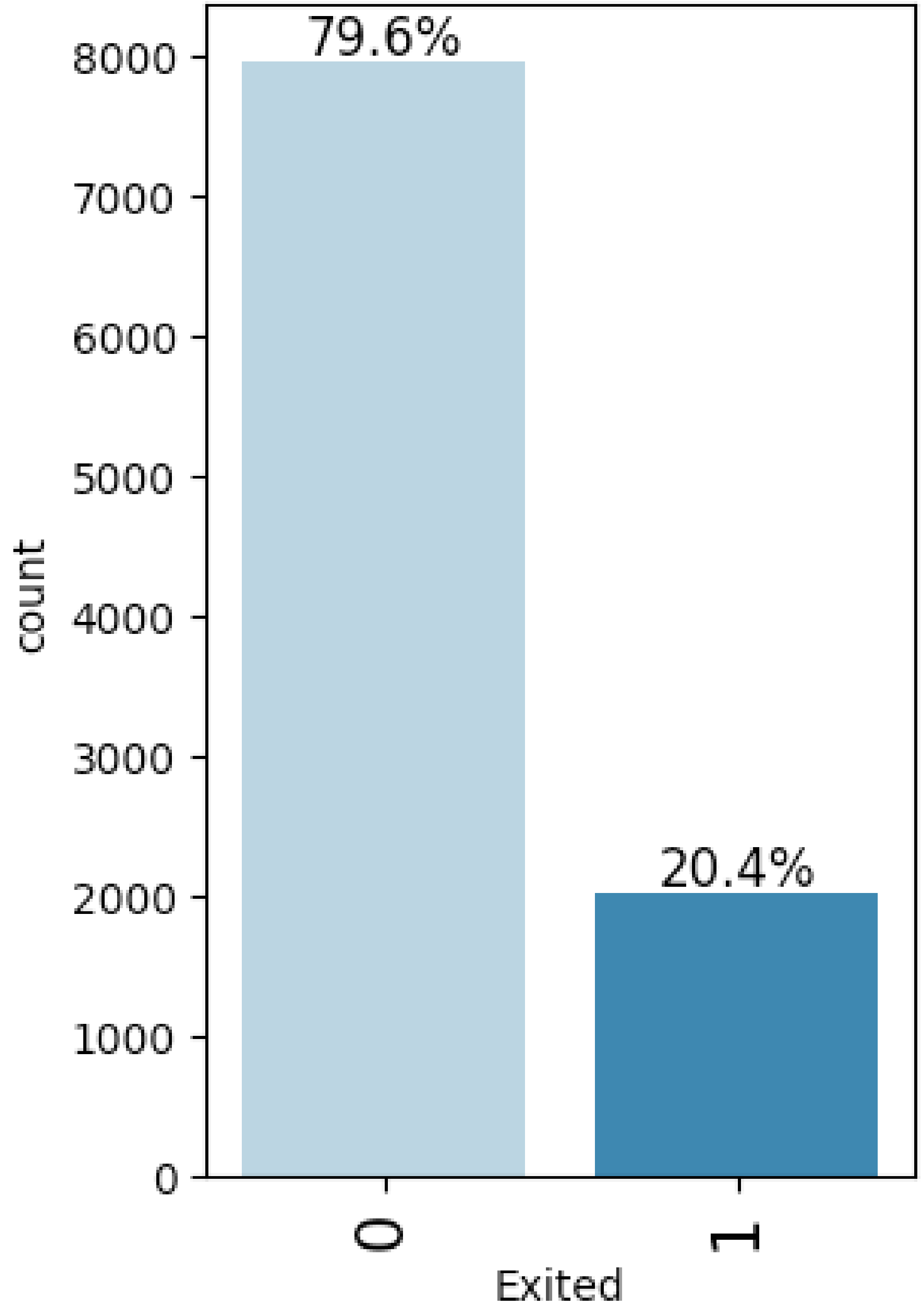
The average account balance for our customers is $76,485.89, with a high standard deviation of $62,397.40, signaling a large variability in balances. Notably, 25% of customers have a zero balance. The median balance is $97,198.54, and the third quartile is at $127,644.20. The highest balance recorded is $250,898.09.
No outliers observed.

EDA Results: Estimated Salary



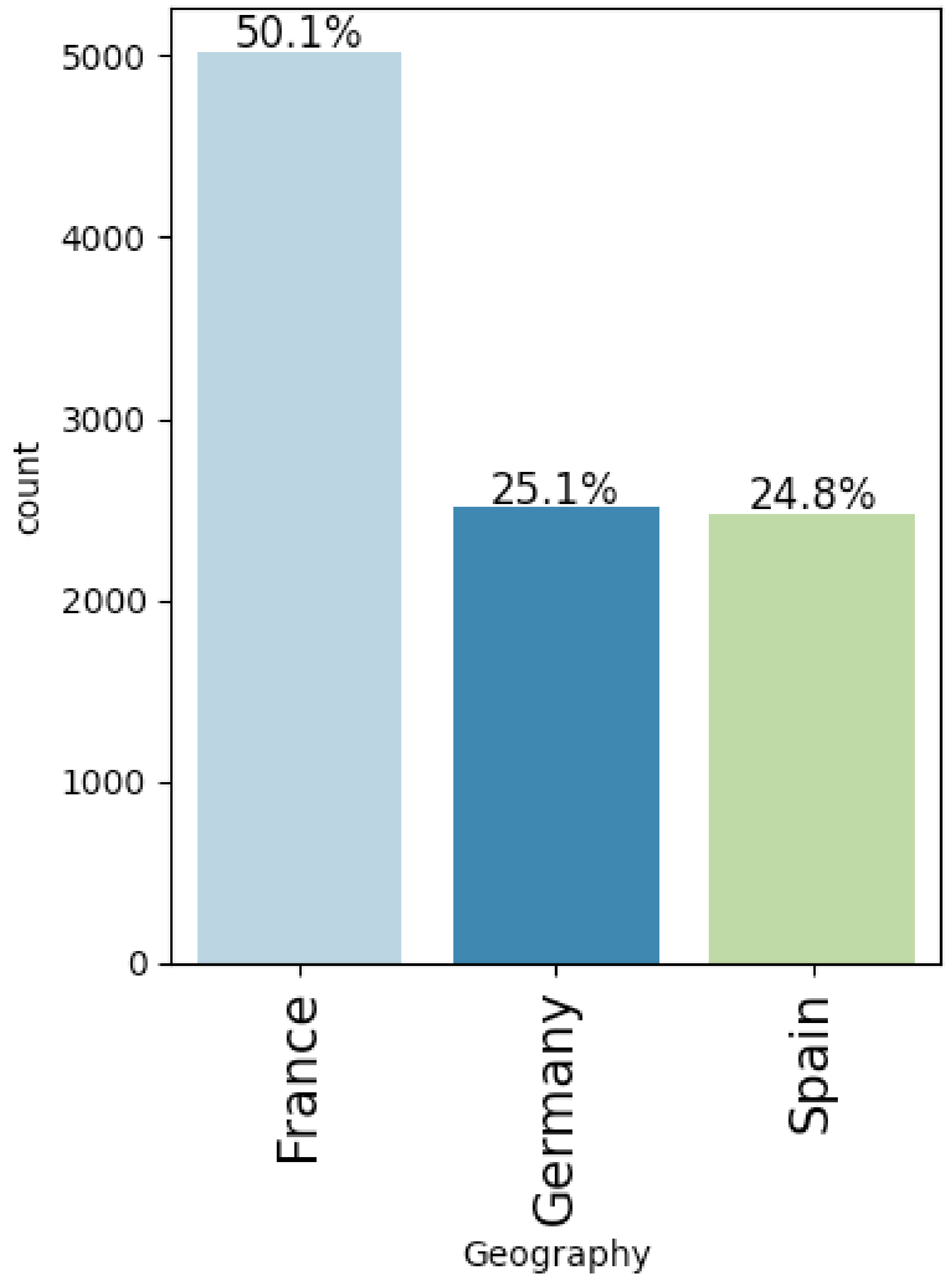
The average estimated salary is $100,090, with a standard deviation of $57,510, suggesting a wide salary range among customers. The salary distribution spans from a minimum of $11.58 to a maximum of $199,992.48.

EDA Results: Exited



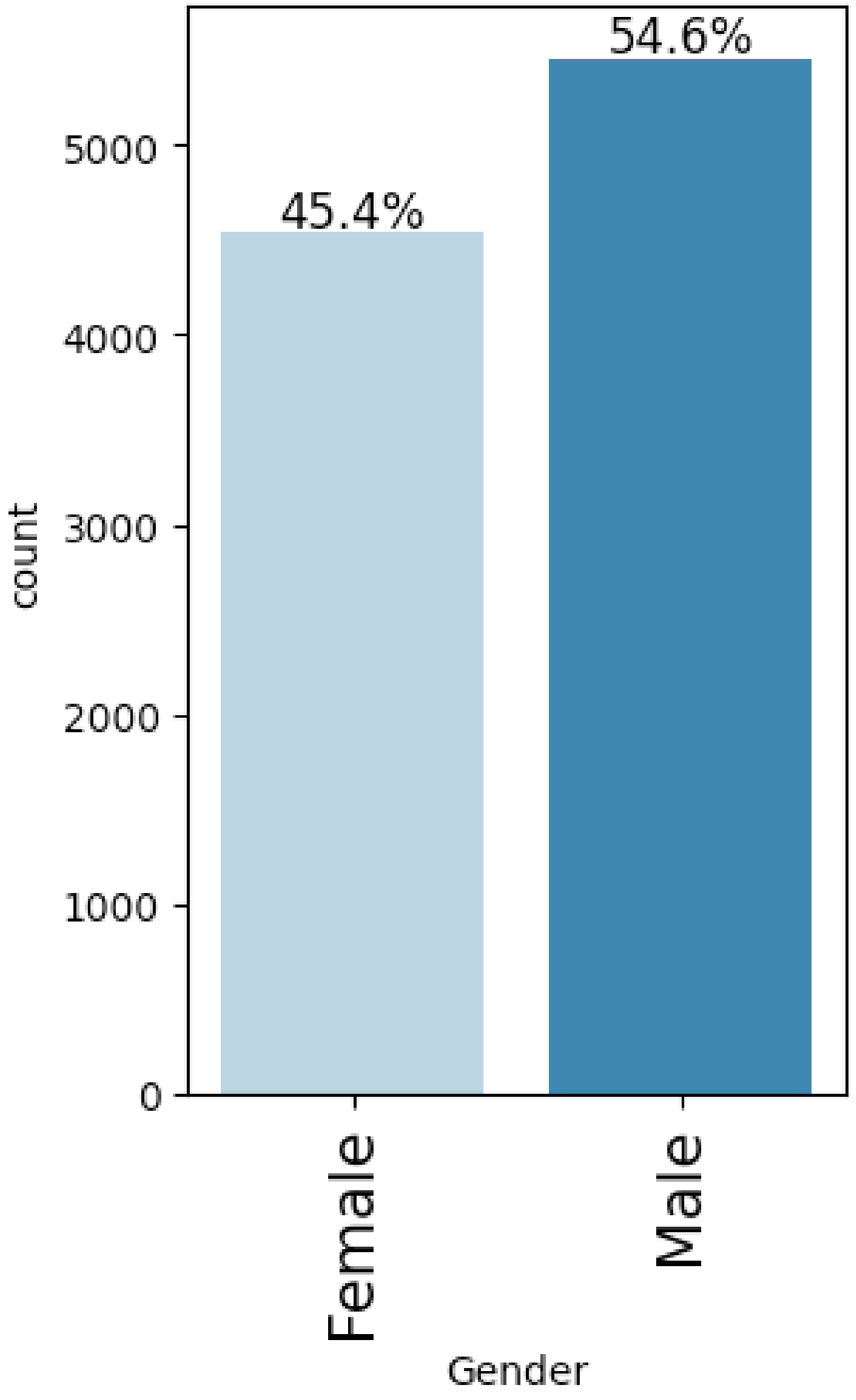
The churn rate, denoted by an average of around 0.20, indicates that around 20% of customers have left the bank. The standard deviation is 0.40. Interestingly, over 75% of the customer base has stayed, signaling a relatively lower churn rate overall and an imbalanced dataset.

EDA Results: Geography



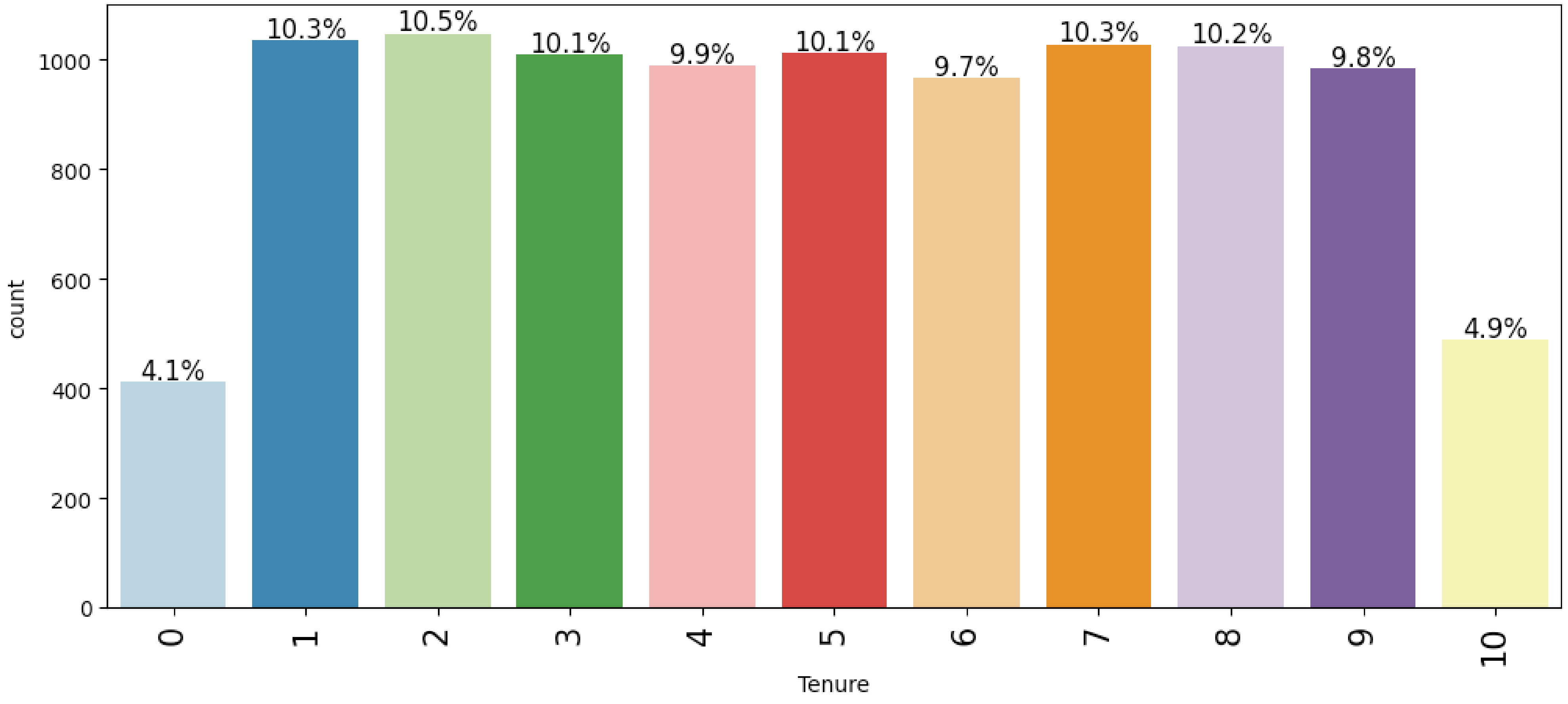
In terms of geography, our customer base is predominantly from France, accounting for 50.1% of the total. Germany and Spain follow with 25.1% and 24.8%, respectively. This geographical spread can be crucial for understanding market-specific behaviors and preferences, as well as for tailoring localized strategies.

EDA Results: Gender



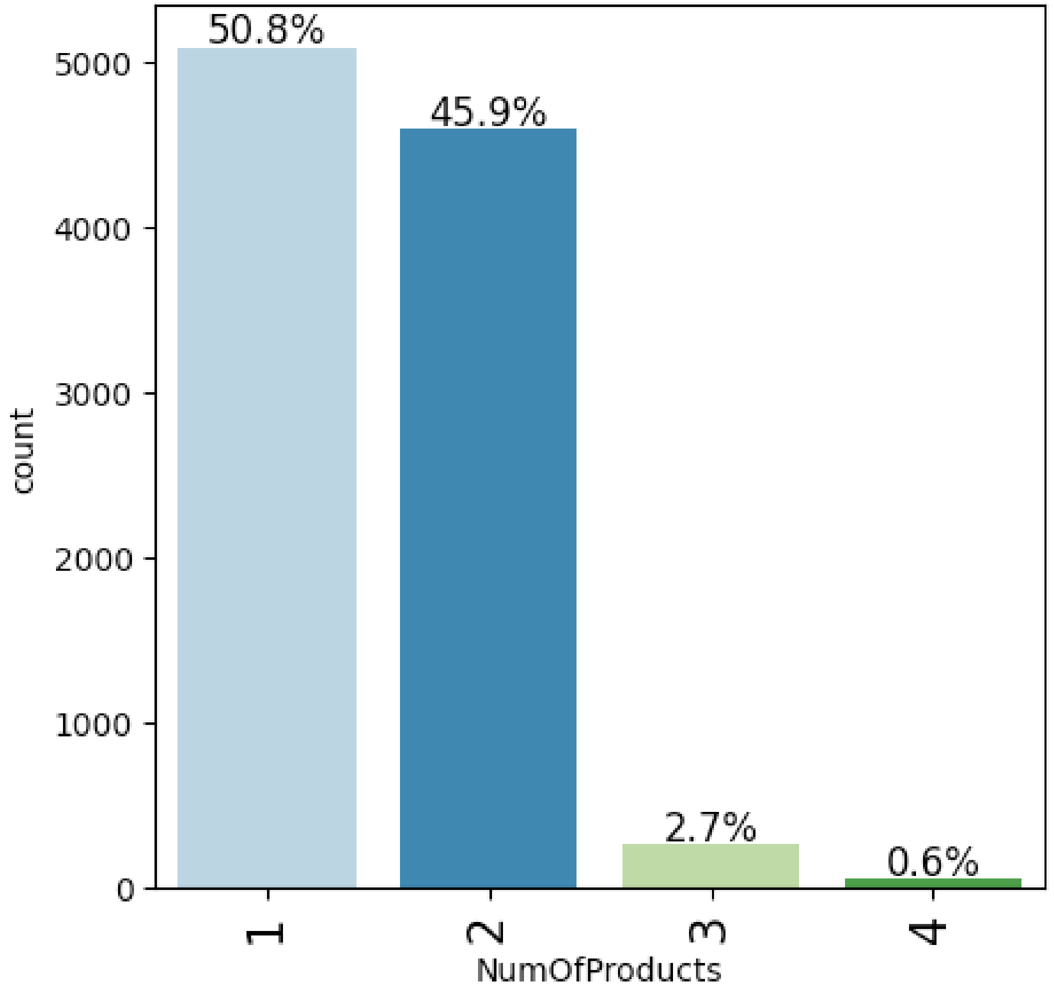
Regarding gender distribution, the dataset shows a slight male dominance with 54.6% being male and 45.4% female. While the gender gap is not substantial, this distribution could be considered in gender-specific marketing campaigns or services.

EDA Results: Tenure



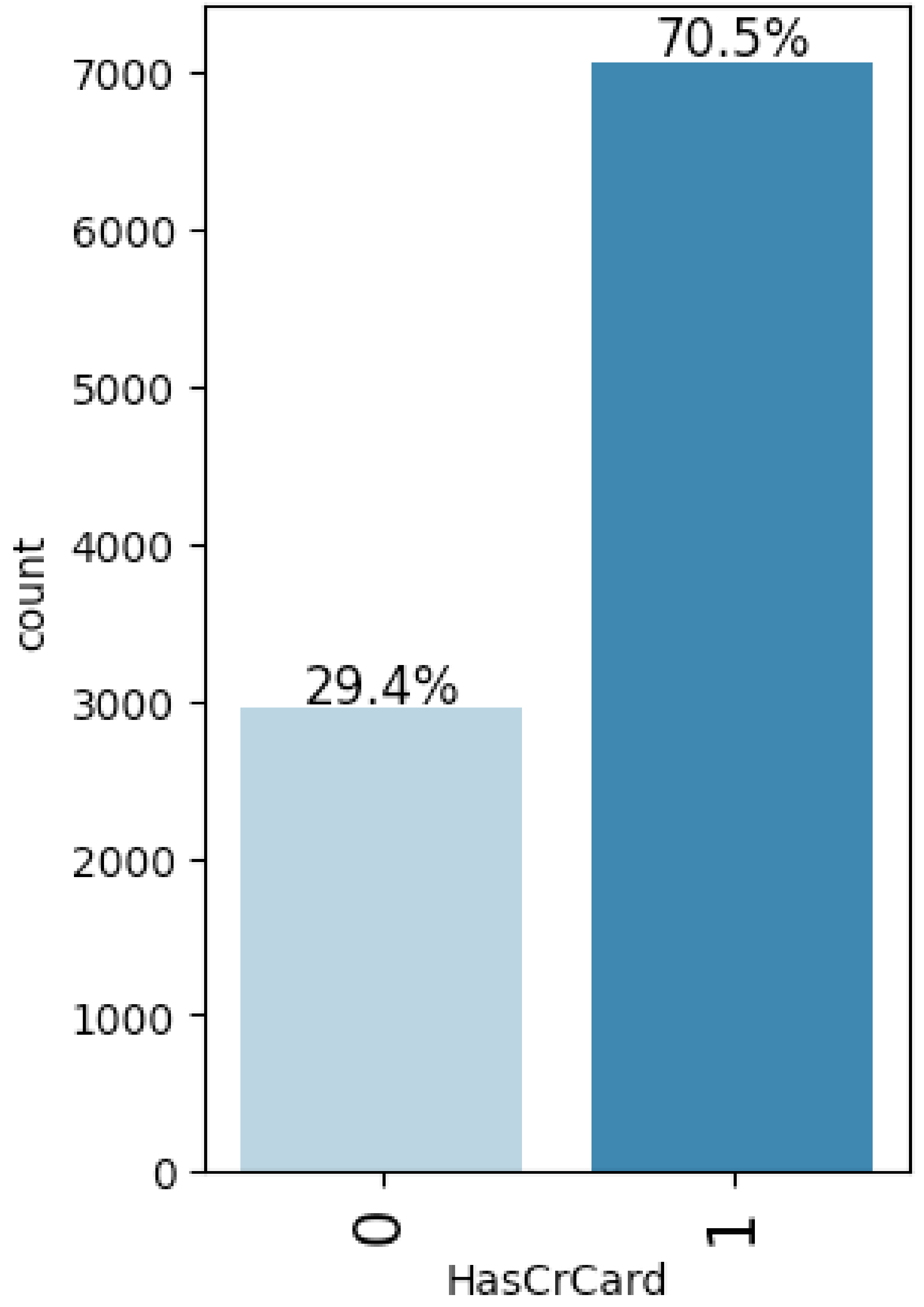
On average, customers have been with the bank for 5 years. The standard deviation is 2.89 years, showing a relatively wide dispersion in customer tenure. New customers, denoted by a tenure of 0 years, make up the minimum. The first quartile is at 3 years, the median is at 5 years, and the third quartile is at 7 years. The maximum tenure observed is 10 years.

EDA Results: Number of Products



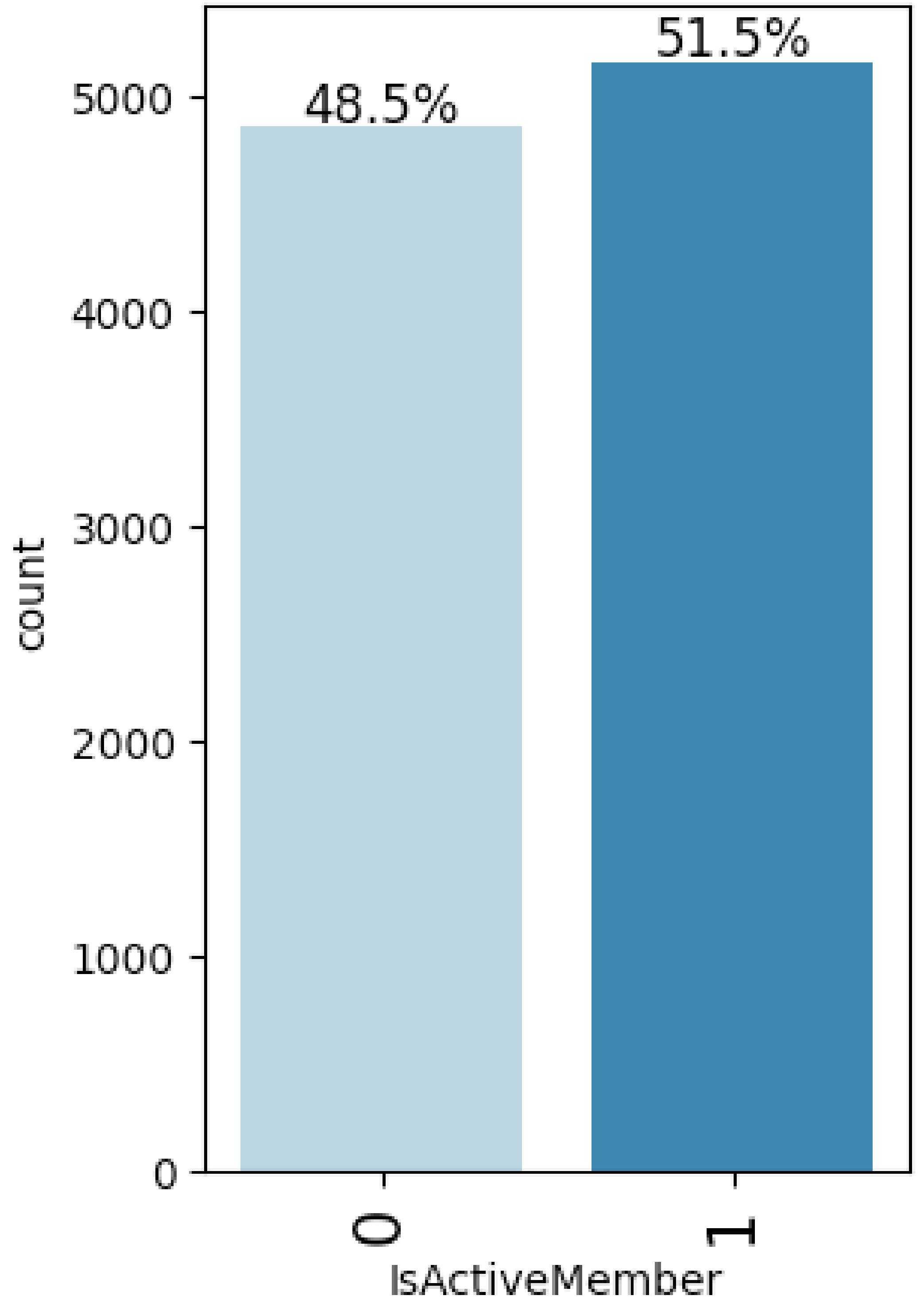
Customers hold an average of 1.53 banking products. The standard deviation is 0.5816, which may seem to suggest limited variability but we need to take on account that there only 4 unique values 1, 2, 3 and 4.
Most customers hold only one product, as evidenced by the first and second quartiles. The third quartile reveals that 75% of customers have up to two products, while the maximum number of products held by any customer is four. 96.7% of the customers only hold up to 2 cards.

EDA Results: Has Credit Card



This is a categorical variable, with an average of 0.71, indicating that a majority of customers own a credit card. The minimum is 0, signifying some customers do not own a credit card, but more than 50% do.

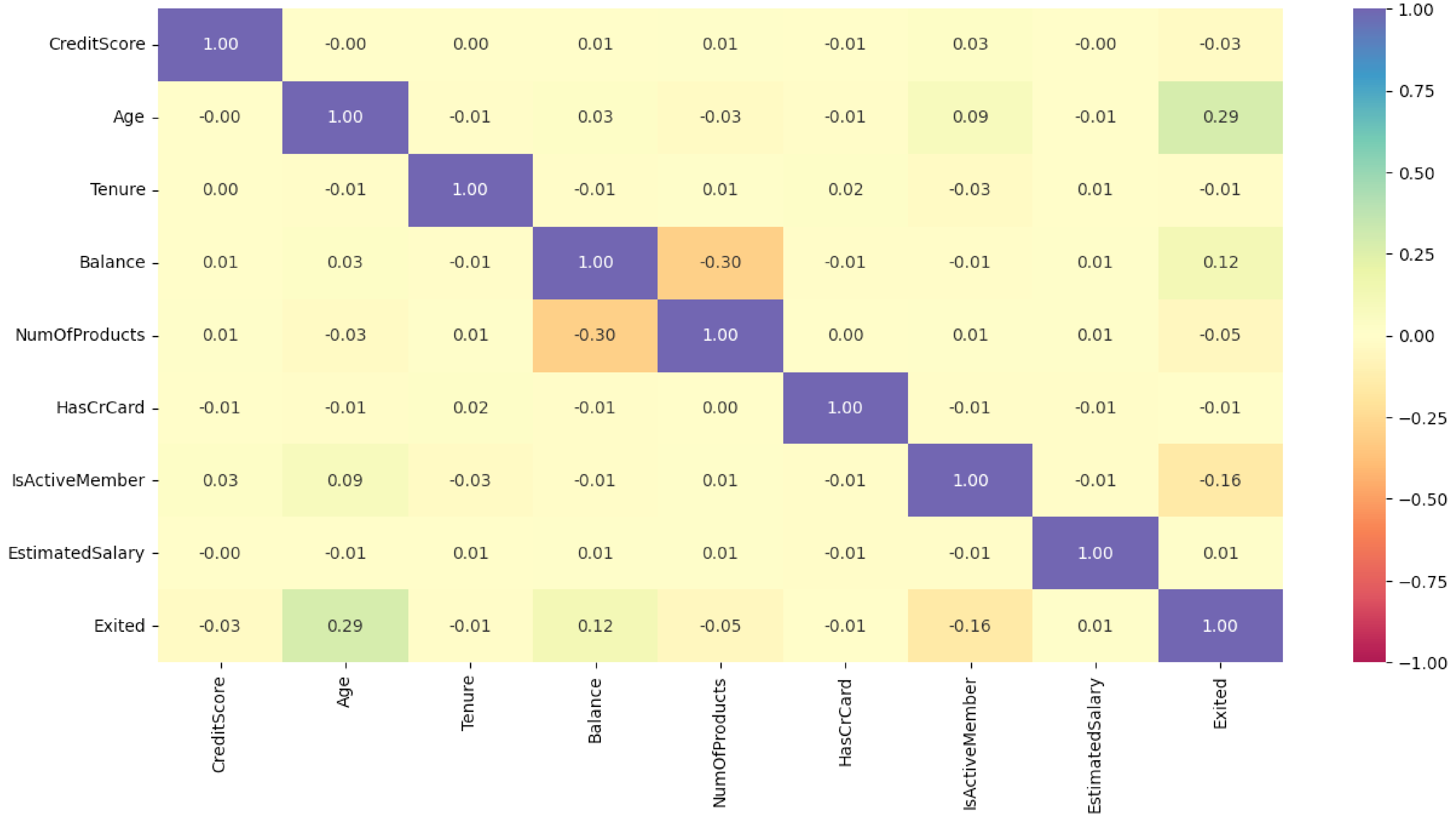
EDA Results: Is Active Member



Approximately 52% of customers are active members, pointing towards a fairly engaged customer base.

This is a a subject of a whole analysis as we need to keep customers active, not only to prevent potential churn but to generate earnings by cards usage, fees, interests, etc.

EDA Results: Correlation Plot (Heatmap)



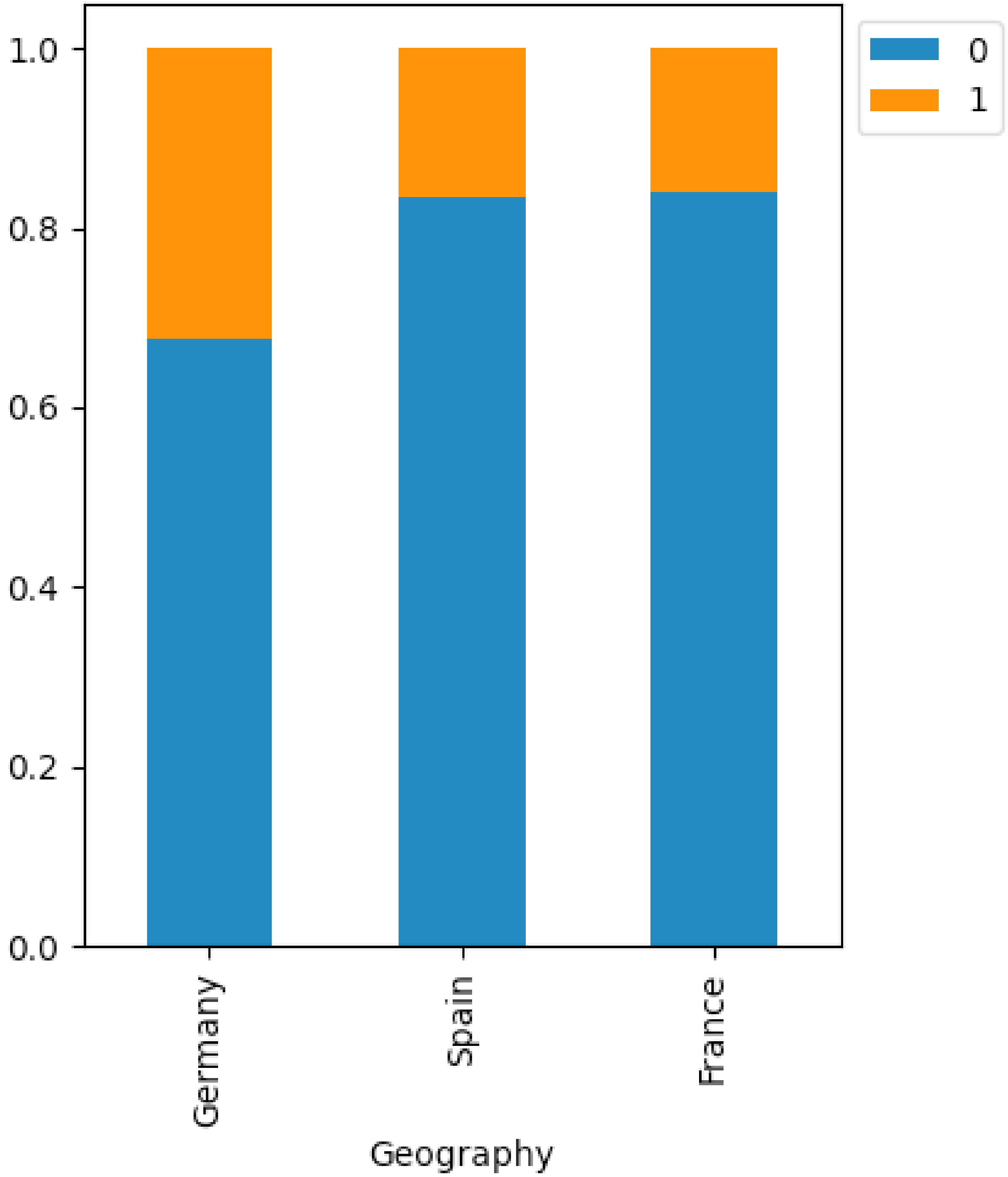
The heatmap indicates a sparse correlation landscape, with only a few notable relationships worth mentioning.

EDA Results: Correlation Plot (Heatmap)

- Age and Exited: There is a positive correlation of 0.29 between age and the likelihood of exiting the bank. This suggests that older customers are more likely to churn.
- Balance and Exited: A positive but weak correlation of 0.12 is observed, indicating that customers with higher balances are slightly more likely to exit.
- Balance and Number of Products: A negative correlation of -0.30 suggests that the more products a customer has, the lower their average balance tends to be. This is intuitive, as a diversified product portfolio may lead to a distribution of funds across various accounts or investments.
- Active Member and Exited: A negative correlation of -0.16 indicates that less active members are slightly more likely to exit the service.

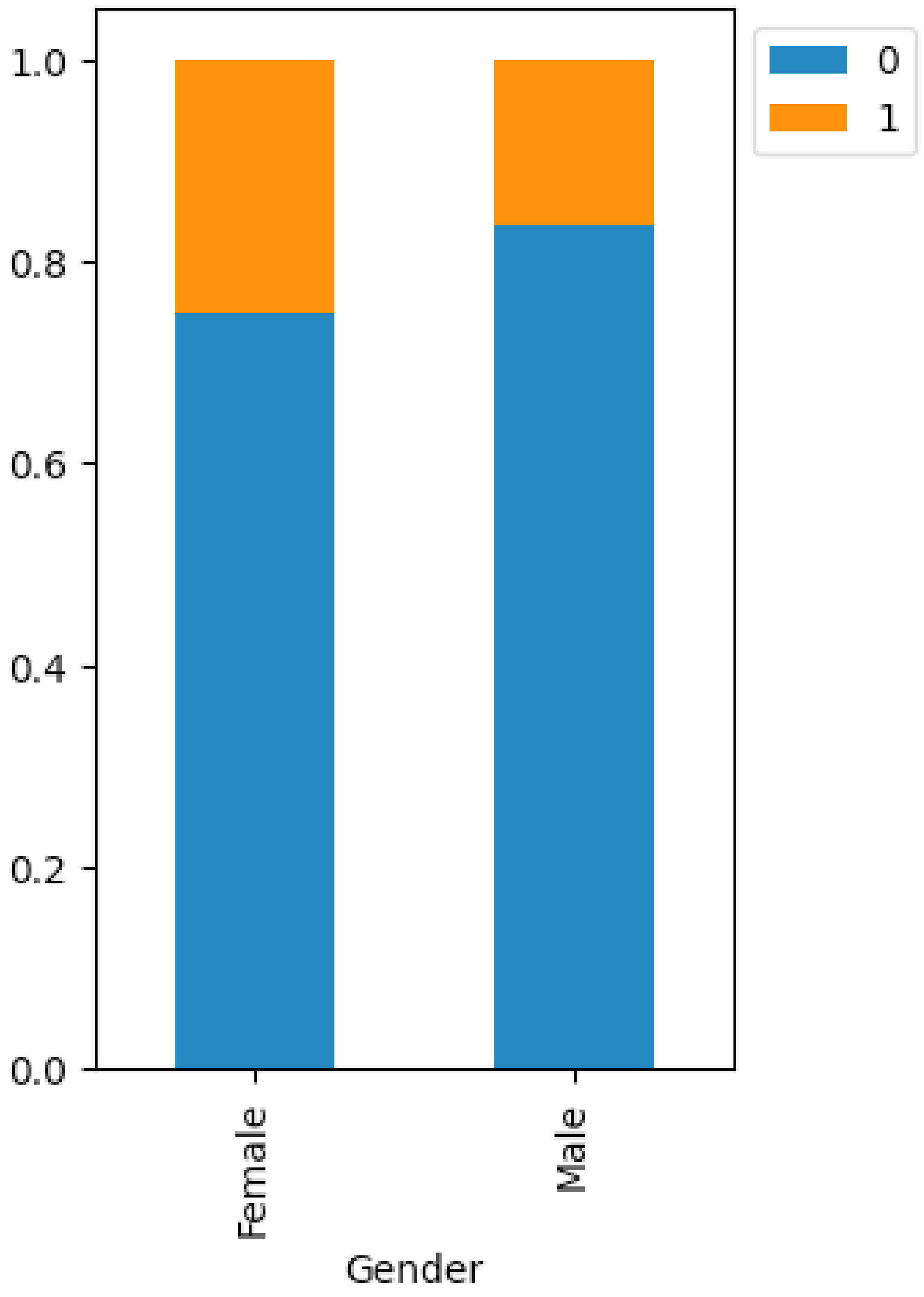
These correlations, while not overwhelmingly strong, offer some initial insights into factors that could influence customer churn and can guide further analysis or feature engineering.

EDA Results: Geography and Exited



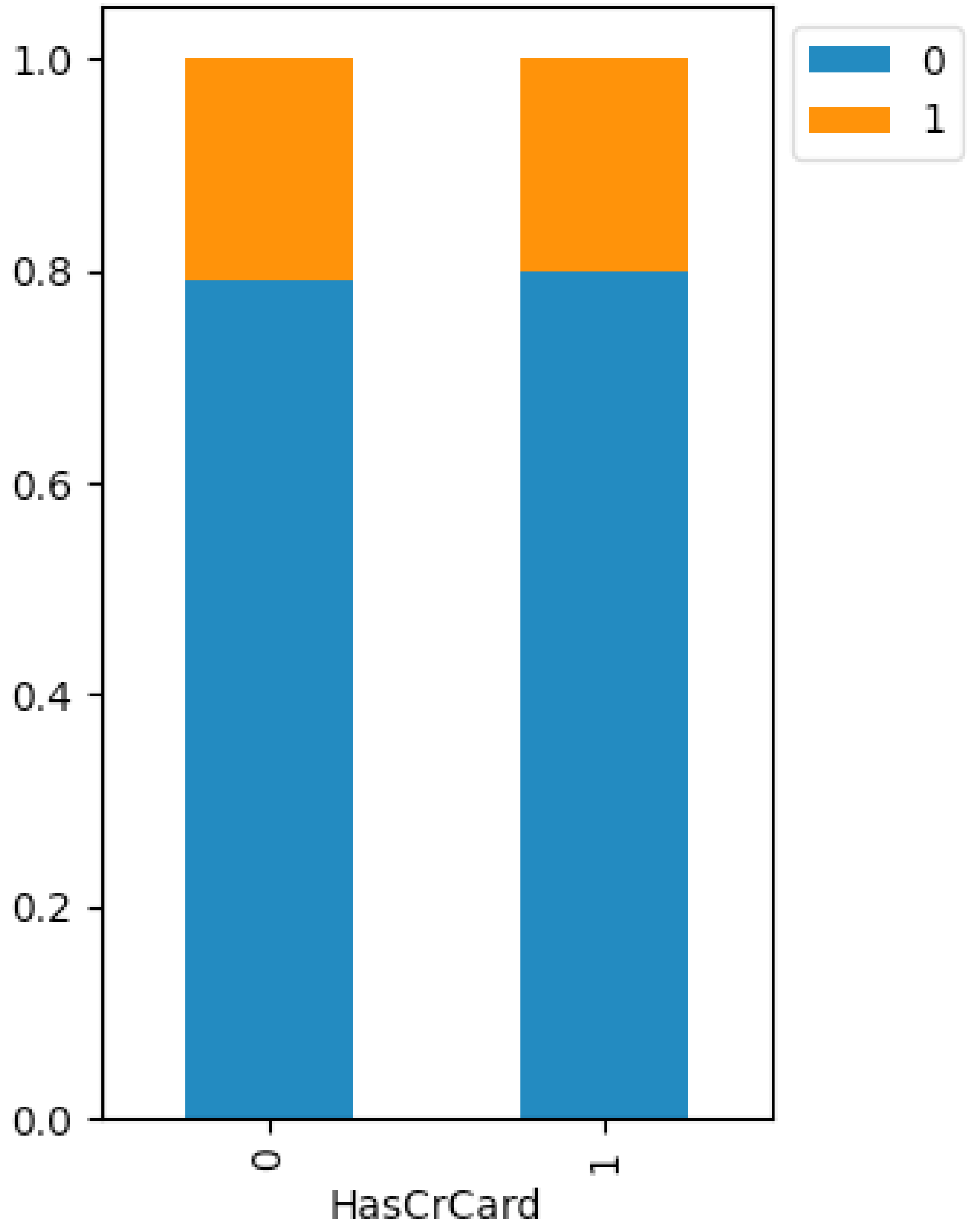
Customers in Germany show a significantly higher churn rate, nearly 35%, as compared to those in Spain and France, where the exit rate hovers near below 20%.

EDA Results: Gender and Exited



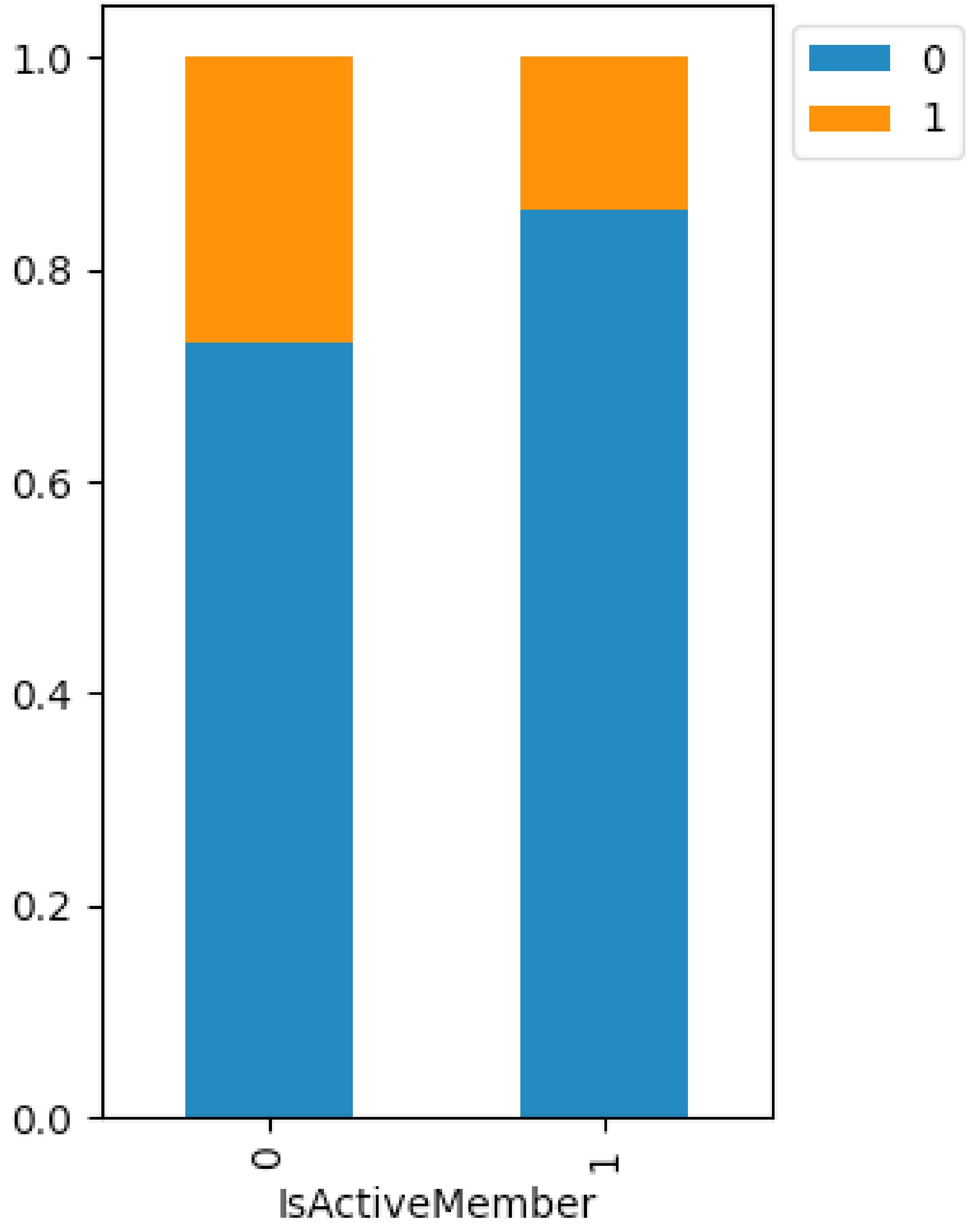
Women exhibit a slightly higher churn rate, slightly above 20%, compared to men, who are below the 20% churn mark.

EDA Results: Has Credit Card and Exited



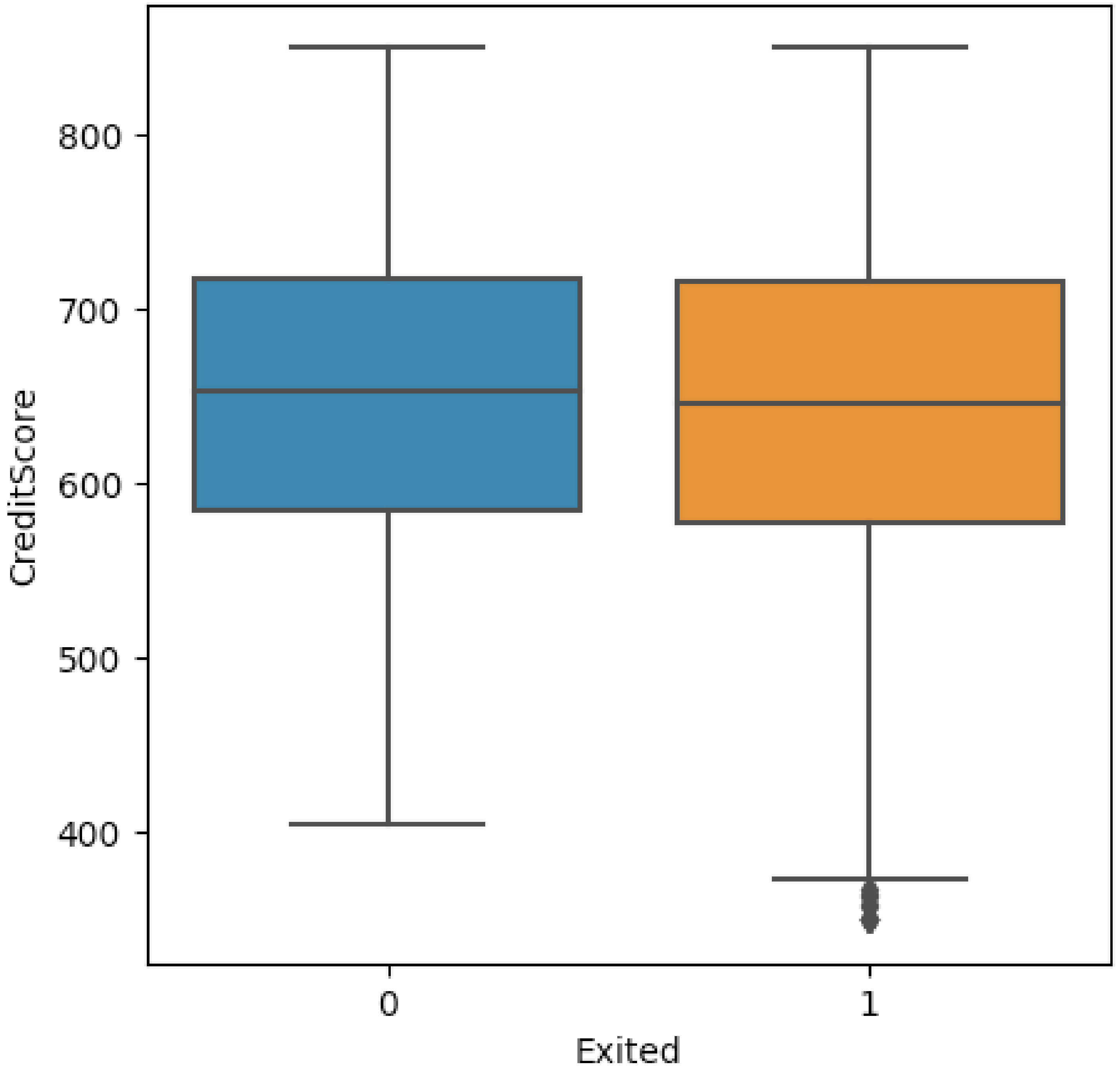
Possession of a credit card doesn't seem to influence the churn rate significantly. Both groups—those with and those without a credit card—have similar exit probabilities.

EDA Results: Is Active Member and Exited



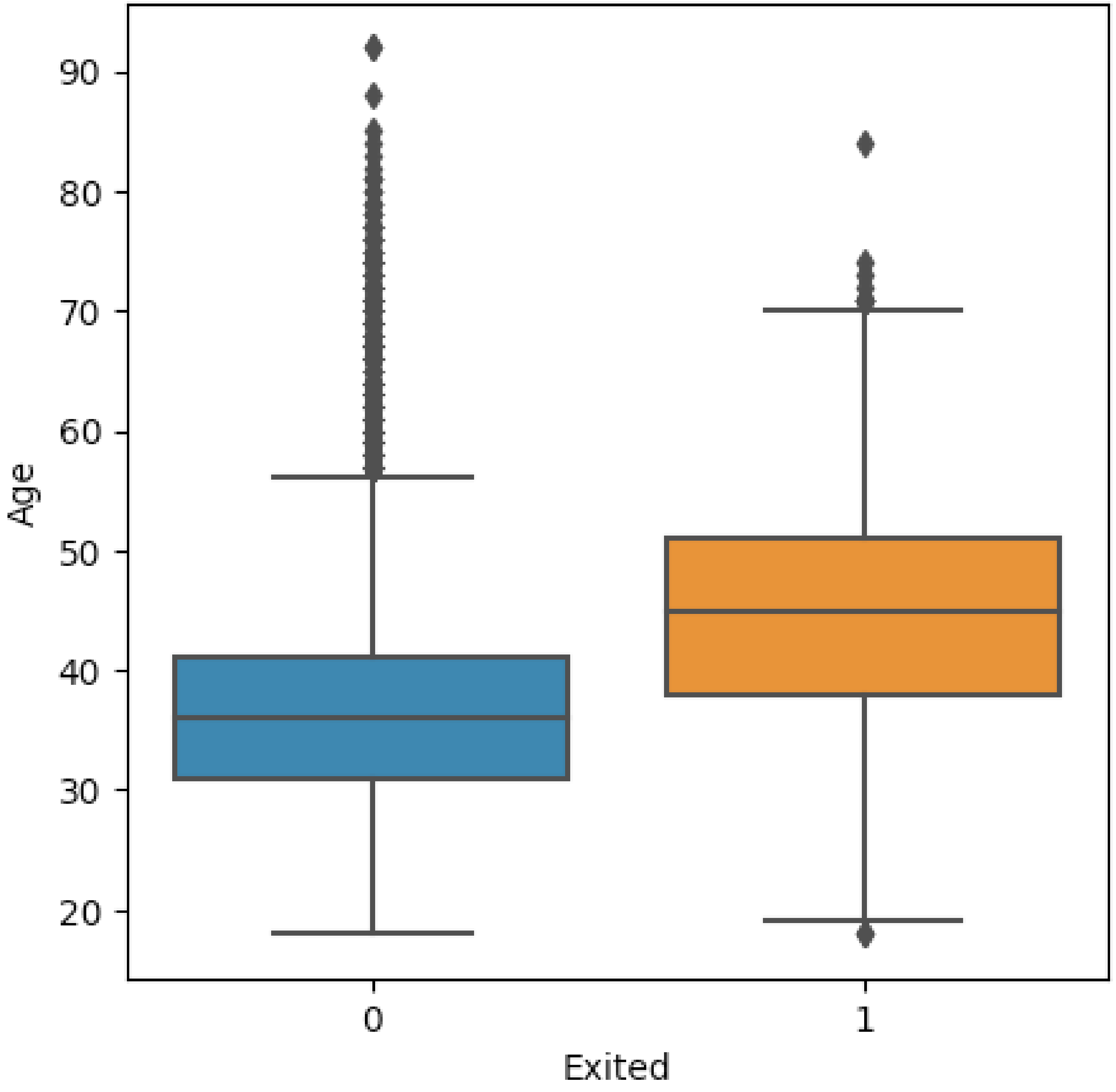
Inactive members show a churn rate above 20%, whereas active members exhibit a lower rate. This reaffirms our earlier findings from the heatmap.

EDA Results: Credit Score and Exited



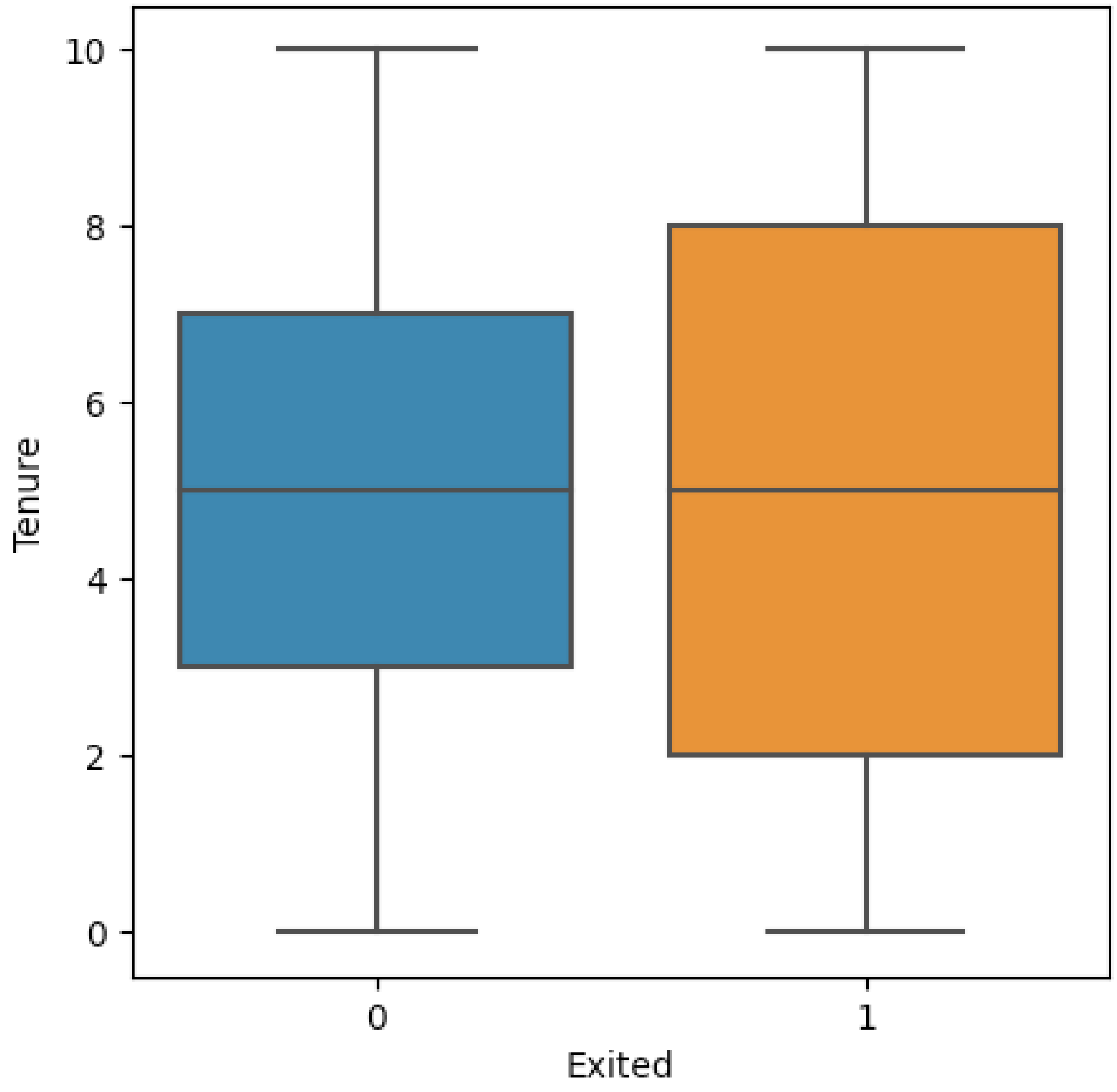
Customers with extremely low credit scores (below 400) are outliers among those who exited. Otherwise, the credit score distribution among those who stayed and those who left appears similar.

EDA Results: Age and Exited



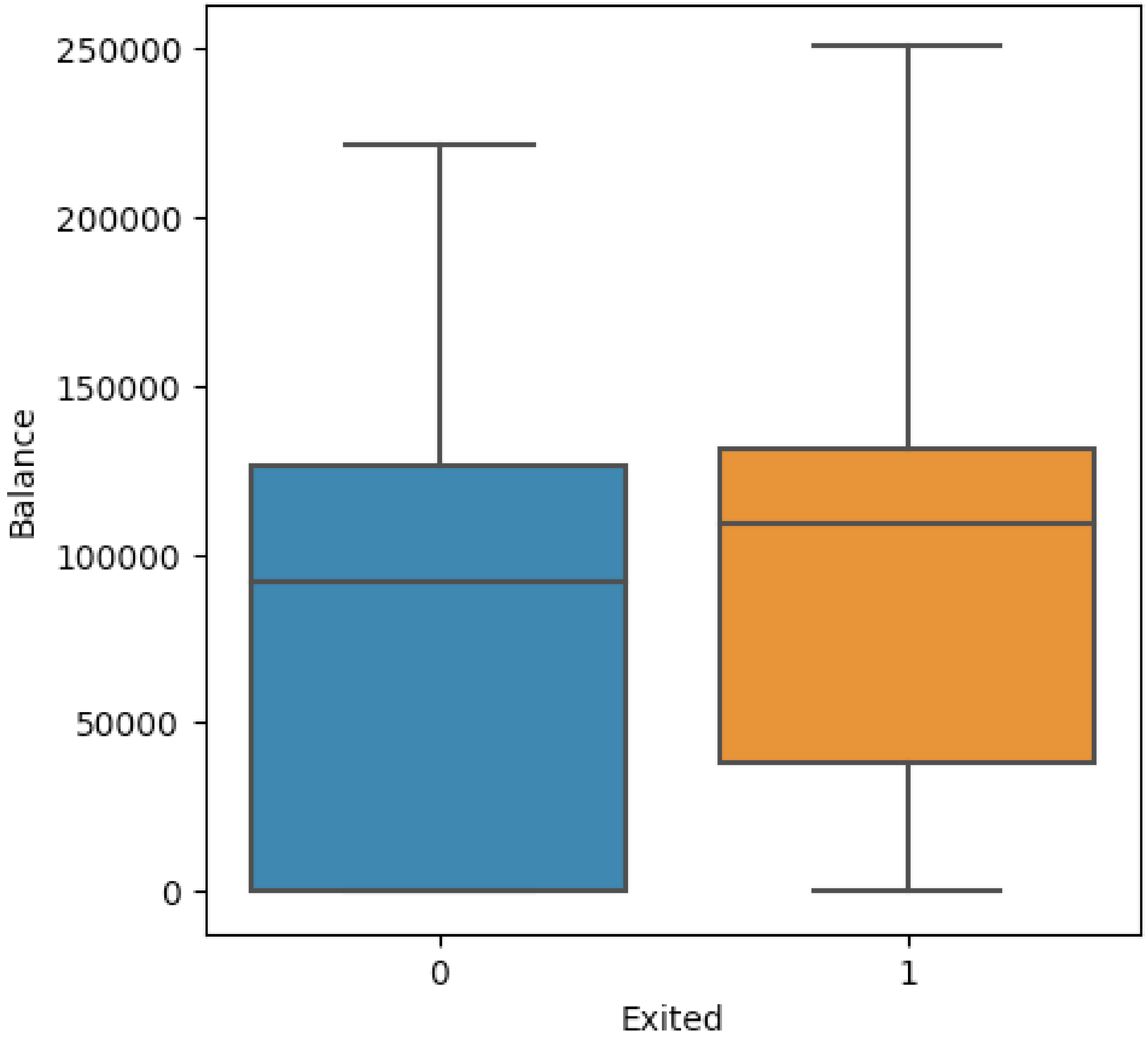
Customers aged between 30 and 40 are more likely to stay, while those over 40 and 50 are more likely to leave. Customers older than 60 are considered outliers among those who stayed, and those older than 70 are outliers among those who left.

EDA Results: Tenure and Exited



The churned customers demonstrate a broader range of tenures, mostly concentrated between 2 to 8 years.

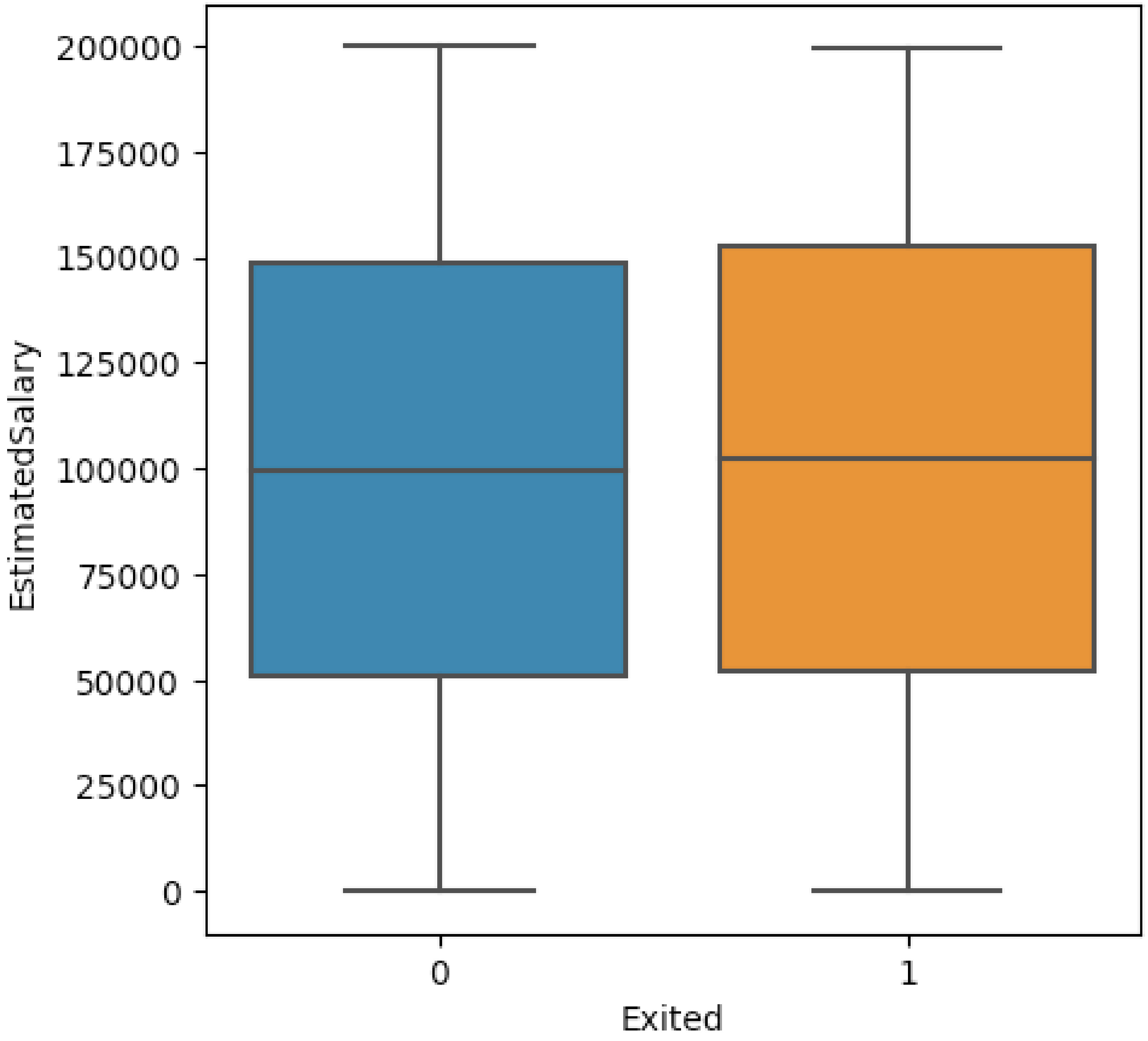
EDA Results: Balance and Exited



Customers who exited generally have higher balances, up to $250,000, compared to those who stayed.

This could mean that some customers have found a service that they feel are better suited for their economic level.

EDA Results: Estimated Salary and Exited



The estimated salary shows no significant difference between those who stayed and those who left, which is consistent with the low correlation observed earlier.

Data Preprocessing

In the Data Pre-Processing phase, we verified the dataset for data quality. No duplicate rows or missing values were found, ensuring the integrity of our dataset. While outliers were identified, the decision was made to retain them, given their potential to provide valuable insights into our target variable, "Exited."
For Feature Engineering, we removed columns that do not contribute to the model's predictive power: 'Row Number,' 'Customer ID,' and 'Surname.' These columns contain unique identifiers and are not useful for our analysis. We then isolated the feature matrix X by removing the 'Exited' column, which was designated as our target variable y.

Data Preprocessing

After isolating the feature matrix X and target variable y, we proceeded with splitting the dataset. Initially, the dataset was split into a larger training set (X_large, y_large) and a testing set (X_test, y_test) with an 80-20 ratio. Stratified sampling was used to ensure that the distribution of the target variable, 'Exited', is consistent across both sets.

Subsequently, X_large and y_large were further divided into a final training set (X_train, y_train) and a validation set (X_val, y_val) again with an 80-20 ratio, adhering to the same stratification and random state principles.

For feature encoding, we used one-hot encoding on the 'Geography' and 'Gender' columns. The drop_first=True parameter was applied to avoid the "dummy variable trap," which could potentially introduce multicollinearity into the dataset.

Data Preprocessing

Finally, feature scaling was performed using the StandardScaler on numerical columns including 'CreditScore', 'Age', 'Tenure', 'Balance', and 'EstimatedSalary'. This was done to normalize the range of these features, making them suitable for algorithms that are sensitive to the scale of input variables. The scaler was fitted only on the training set to prevent data leakage, and the same scaling parameters were applied to the test and validation sets.

Model Performance Summary: Classifier Model

The input layer will have 64 neurons with ReLU (Rectified Linear Unit) as the activation function. The input dimension is 11, which is based on the number of features we've decided to include in our model.

The first hidden layer will have 32 neurons, and we've chosen to use ReLU as the activation function.

The output layer will have just one node because we're dealing with binary classification: whether the customer will churn or not. The activation function to use here is 'sigmoid', as it will squash the output between 0 and 1, which is what we desire for a binary classification problem.

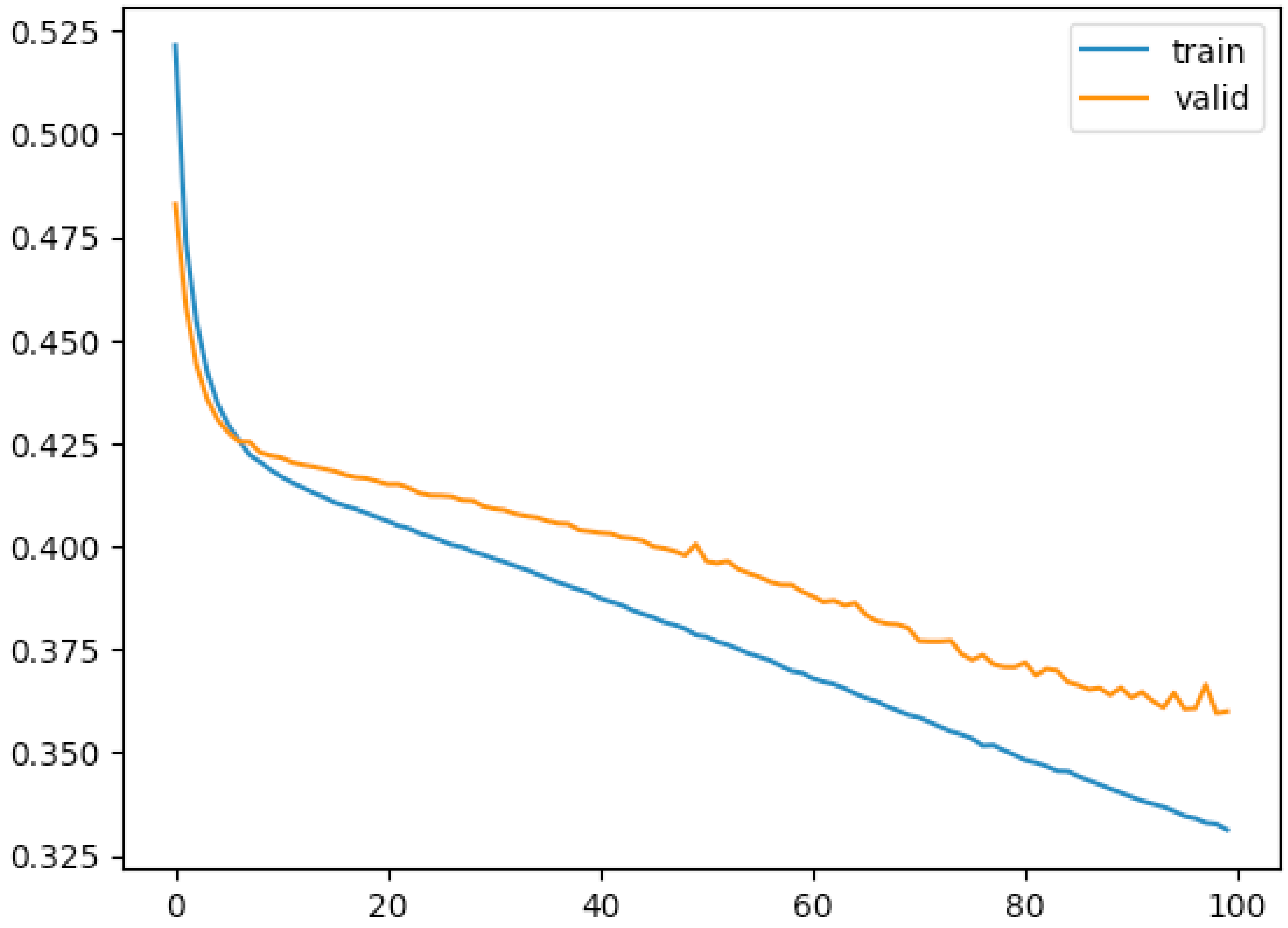
Model Performance Summary: Classifier Model

1. Optimizer: We are using the Stochastic Gradient Descent (SGD) optimizer. The optimizer's role is to adjust the weights in the neural network to minimize the loss function.
2. Loss Function: Since this is a binary classification problem, the appropriate loss function is Binary Cross-Entropy.
3. Metrics: You want to track accuracy as the metric for this problem, which is quite common for classification problems.

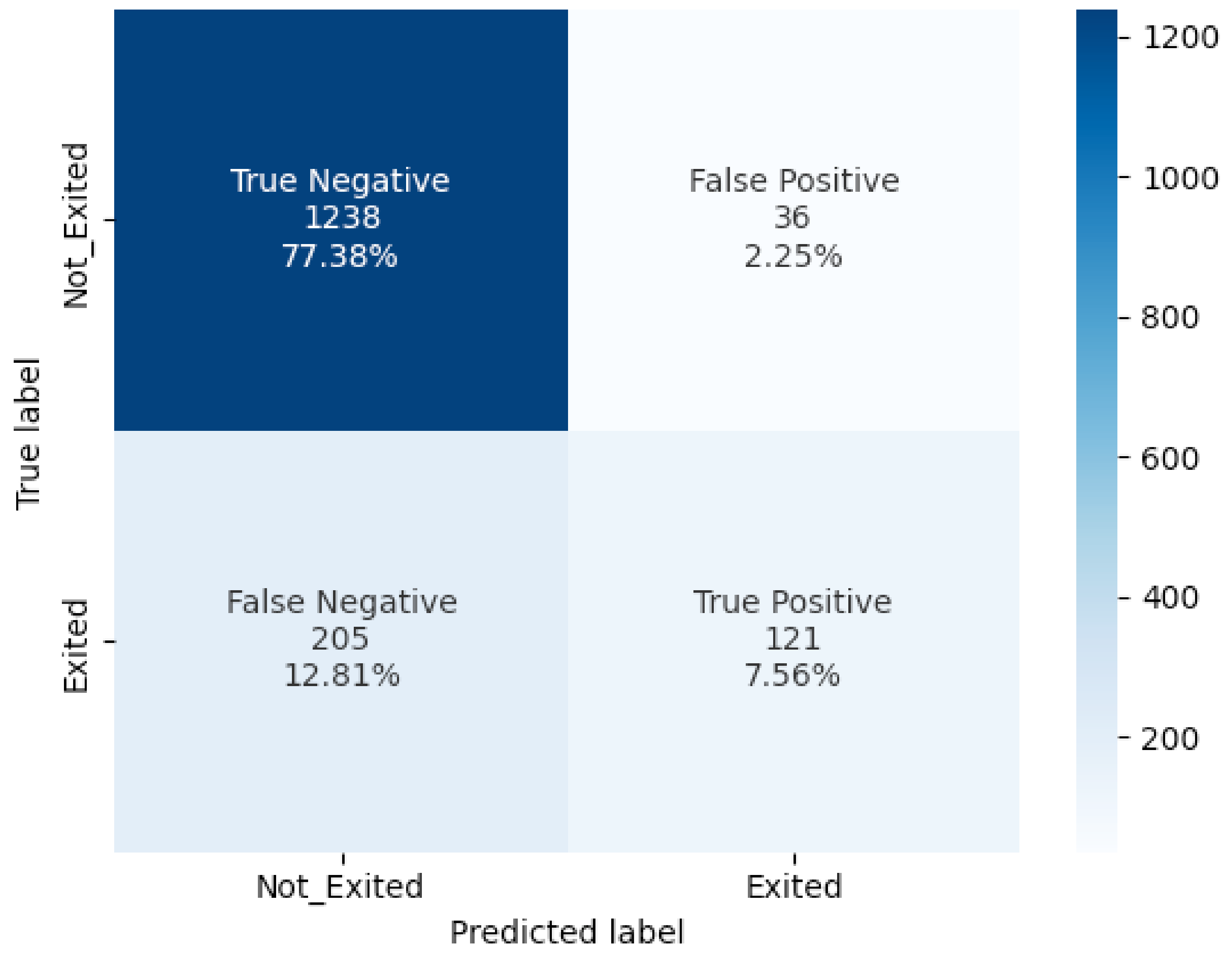
1. For training we used 100 Epochs: This number of epochs should be enough for the model to converge and Batch Size 32 which is a commonly used batch size and generally provides a good balance between speed and accuracy of the weight updates.

Model Performance Summary: Classifier Model

We see no overfitting in the plot but there is room for improvement in the loss function.



Model Performance Summary: Classifier Model



Accuracy=0.849
Precision=0.771
Recall=0.371
F1 Score=0.501

The confusion matrix for the first classifier model shows that True Negatives account for 77.38%, indicating that the model is good at identifying customers who are not likely to churn. However, there's room for improvement in the area of False Negatives, which stand at 12.81%.

False Negatives are particularly concerning for a churn prediction model because they represent missed opportunities to retain customers; these are the customers who are predicted to stay but actually leave.

Model Performance Summary: Classifier Model

The Accuracy of 0.849 suggests that the model generally classifies the data well, but this metric can be misleading in imbalanced datasets. The Precision score of 0.771 implies that among the customers the model predicted would churn, 77.1% actually did. This is relatively good but could be improved.

The most concerning metric is Recall at 0.371, which is quite low. Recall tells us how well the model is identifying all the possible churn cases. A low Recall means the model is missing out on a significant number of actual churn cases, which is not ideal in a customer churn setting.

The F1 Score, which is the harmonic mean of Precision and Recall, stands at 0.501. This score suggests that the model is not balanced and leans more towards Precision. Improving the Recall without substantially sacrificing Precision should be a focus for model improvement.

Model Performance Summary: Classifier Model with Adam Optimizer

Input Layer: The model starts with a dense layer of 64 neurons with ReLU activation. The input dimension is dynamically set to match the number of features in X_train.

Hidden Layer: A dense hidden layer follows, containing 32 neurons with ReLU activation.

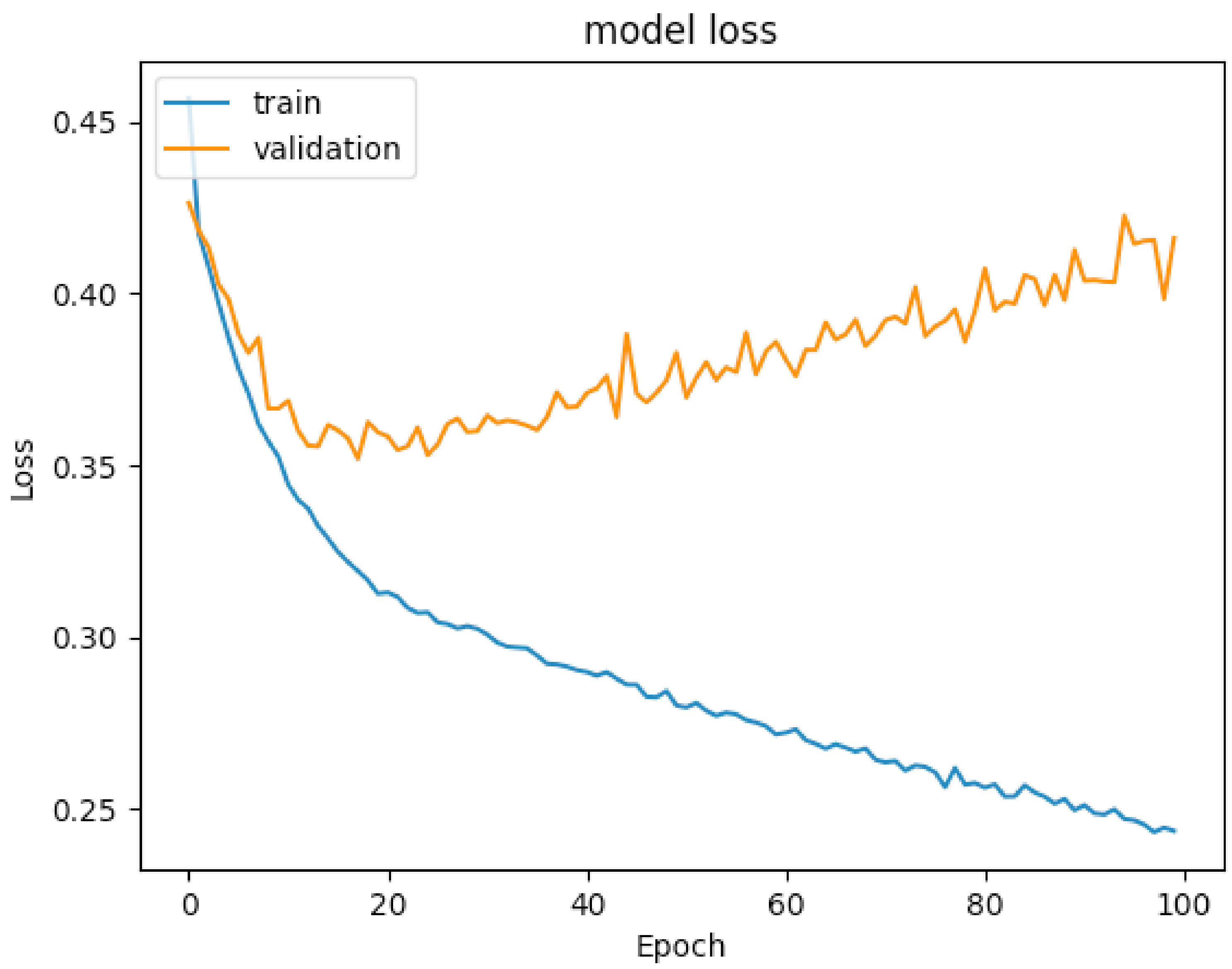
Output Layer: The output layer is a dense layer with 1 neuron, using a sigmoid activation function for binary classification.

Optimizer: You've indicated that you want to use the Adam optimizer, initialized with a learning rate of 0.001.

Compilation: The model will be compiled using binary cross-entropy as the loss function and will track accuracy as a metric.

Model Performance Summary: Classifier Model with Adam Optimizer

We are using 100 epochs and a batch of 32 again, and we can observe a severe overfitting. The best solution for this problem is Early stopping.



Model Performance Summary:
Classifier model_e with Adam Optimizer Early Stopping

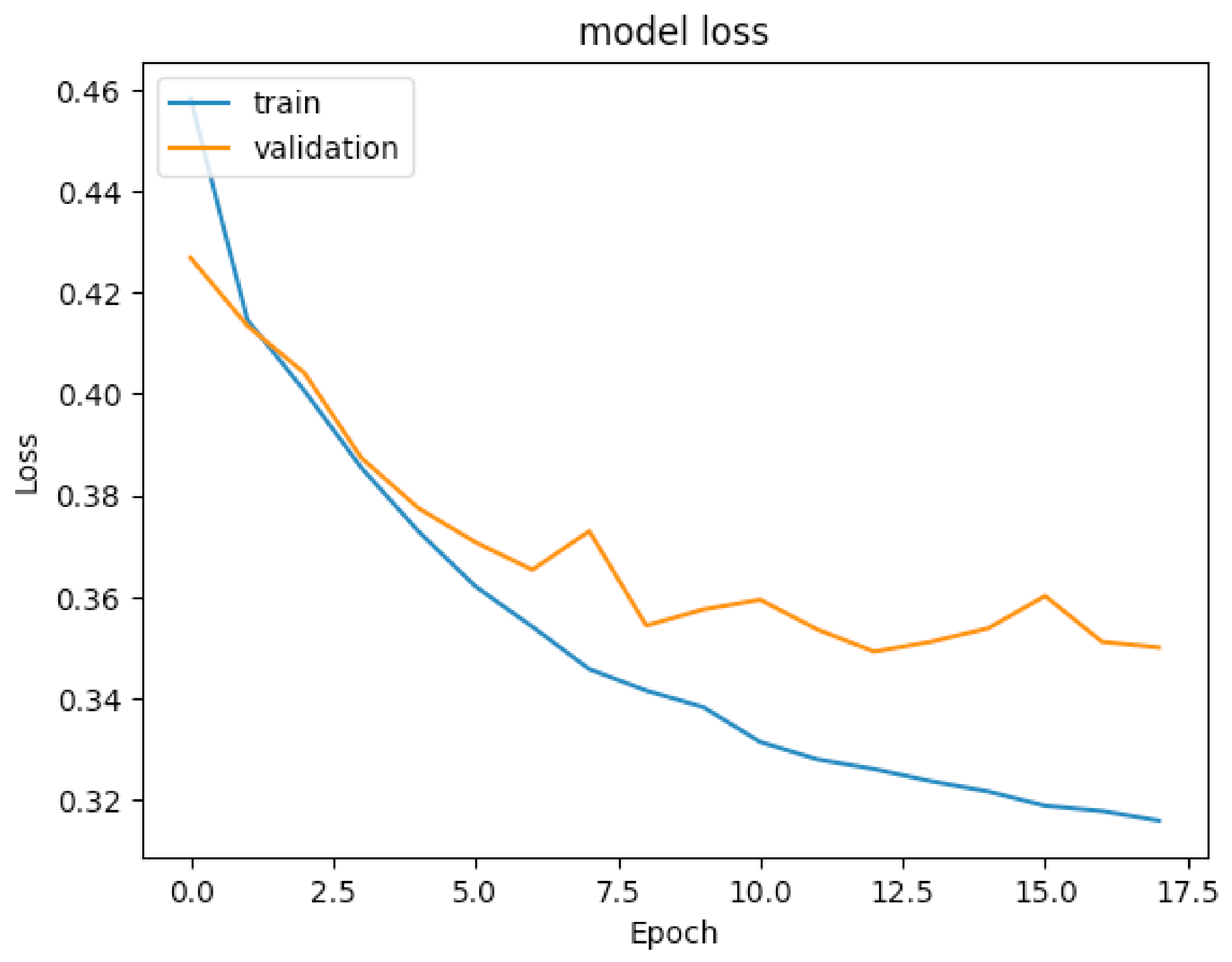
Early Stopping callback is employed to monitor the validation loss during training. The callback will stop the training process if the validation loss does not improve by a minimum delta of 0.001 over five consecutive epochs. This is a regularization technique to prevent overfitting by stopping the training when the model starts to memorize the training data instead of generalizing from it.

With a batch size of 32 and a maximum of 100 epochs. The validation set is also passed to monitor the model's performance on unseen data during training. The Early Stopping callback is added to the training process via the callbacks parameter.

Model Performance Summary: Classifier model_e with Adam Optimizer Early Stopping

No overfitting is shown but the loss function on validaton data could be improved.

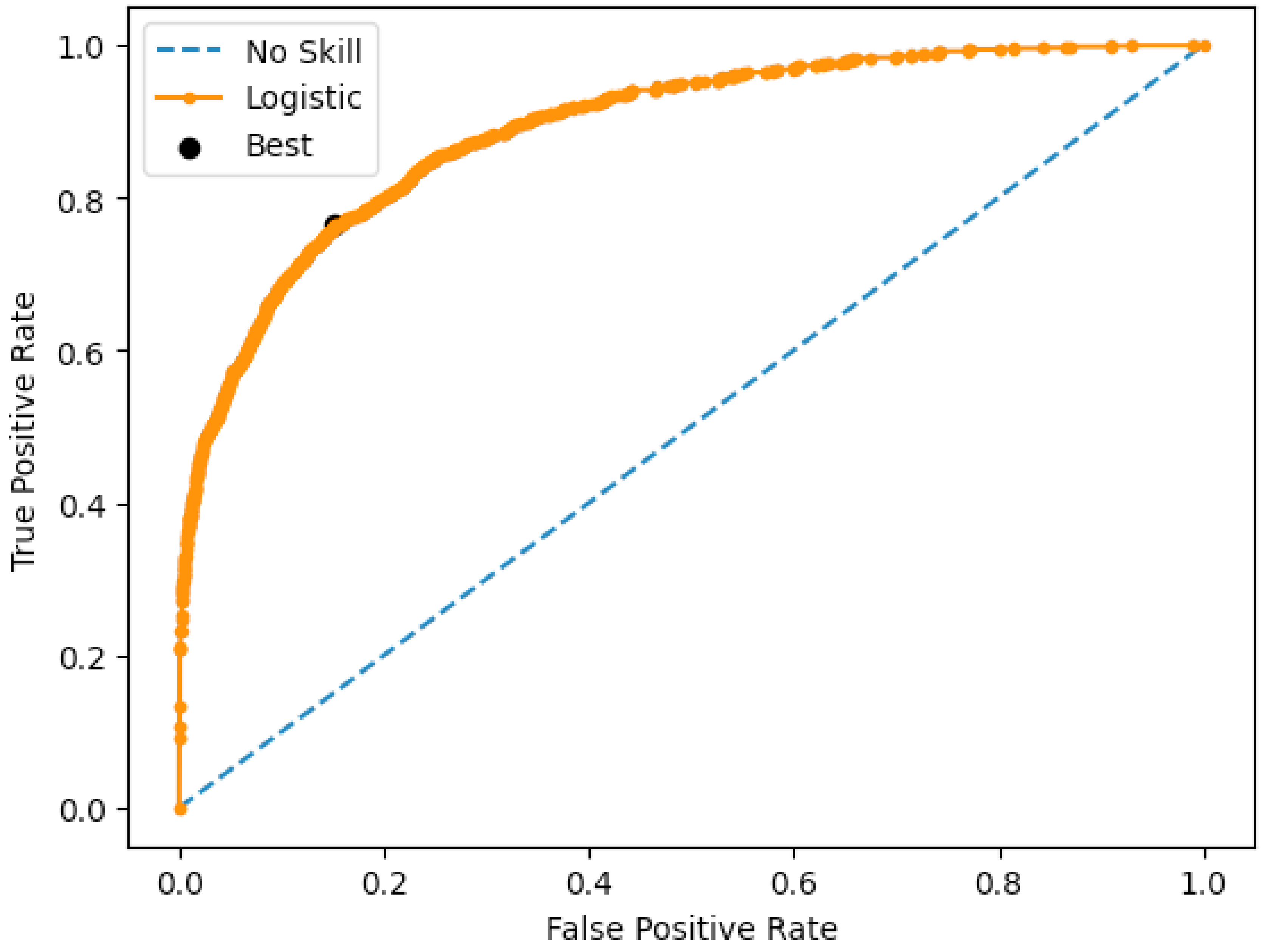
Let's tune the threshold using ROC-AUC



Model Performance Summary: Model_e ROC-AUC Tuning

The ROC curve plots the true positive rate (Sensitivity) against the false positive rate (1 - Specificity) for different threshold values. The point closest to the top-left corner of the plot gives you the best trade-off between Sensitivity and Specificity, or in terms of G-Mean, the best geometric mean of the two.

Model Performance Summary: Model_e ROC-AUC Tuning



Best Threshold = 0.209287: This is the probability threshold at which the G-Mean is maximized. If the model predicts a probability higher than this for a given instance, that instance will be classified as 'Exited'.

G-Mean = 0.805: This value indicates a good balance between Sensitivity and Specificity. The closer the G-Mean is to 1, the better the balance.

Model Performance Summary: Model_e ROC-AUC Tuning

	Precision	Recall	F1-Score	Support
0	0.92	0.85	0.88	1274
1	0.55	0.71	0.62	326
Accuracy			0.82	1600
Macro avg	0.73	0.78	0.75	1600
Weighted avg	0.84	0.82	0.83	1600

Accuracy: The model has an overall accuracy of 82%, indicating that it correctly classified 82% of all examples in the validation set.

Precision and Recall for Class 0 (Not Exited):

Precision is 0.92, indicating that 92% of the instances predicted as 'Not Exited' actually are 'Not Exited'.

Recall is 0.85, indicating that the model correctly identified 85% of all 'Not Exited' instances.

Model Performance Summary: Model_e ROC-AUC Tuning

Precision and Recall for Class 1 (Exited):

Precision is 0.55, which means that 55% of the instances predicted as 'Exited' actually did exit.

Recall is 0.71, indicating that the model was able to correctly identify 71% of all 'Exited' instances.

F1-Score:

For Class 0, the F1-Score is 0.88, which is quite good and suggests that the model is performing well for this class.

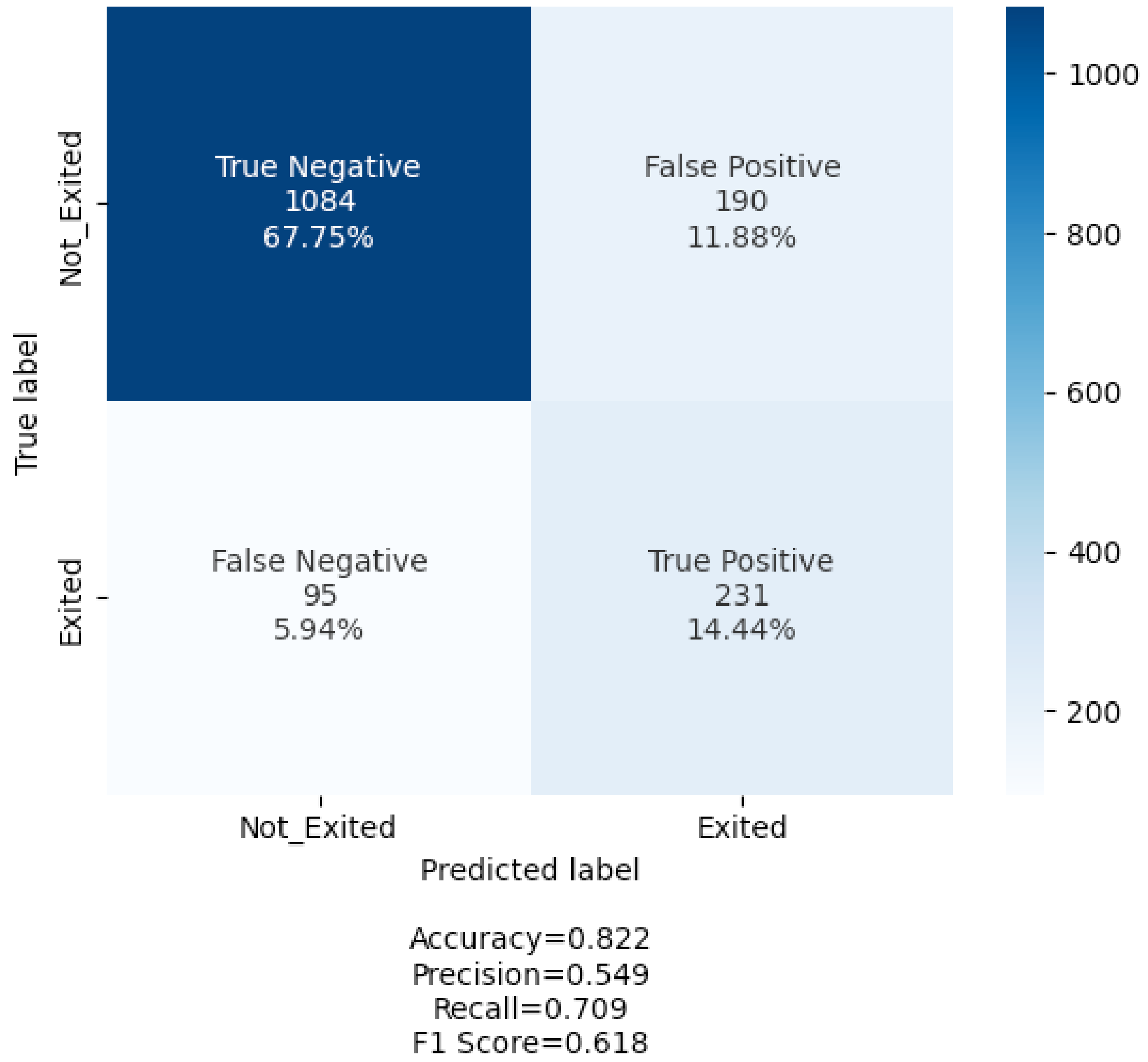
For Class 1, the F1-Score is 0.62, which is moderate and suggests that there is room for improvement for this class.

Model Performance Summary: Model_e ROC-AUC Tuning

Macro Average: The macro-average F1-Score is 0.75, which gives equal weight to both classes and is a reasonable metric if you care about both classes equally.

Weighted Average: The weighted average F1-Score is 0.83, which is weighted by the number of instances for each class and is more indicative if there's a class imbalance.

Model Performance Summary: Model_e ROC-AUC Tuning



True Negatives: 1084 instances were correctly classified as 'Not Exited,' making up 67.75% of the total instances.

False Positives: 190 instances were incorrectly classified as 'Exited,' constituting 11.88% of the total instances.

False Negatives: 95 instances were incorrectly classified as 'Not Exited,' constituting 5.94% of the total instances.

Model Performance Summary: Model_e ROC-AUC Tuning

True Positives: 231 instances were correctly classified as 'Exited,' making up 14.44% of the total instances.

Accuracy: The overall accuracy of the model is 82.2%, which suggests that the model is quite reliable for making general predictions.

Precision for 'Exited': The model has a precision of 0.549, meaning that roughly 55% of the instances predicted as 'Exited' were actually 'Exited.'

Recall for 'Exited': The recall rate is 0.709, indicating that the model correctly identified approximately 71% of all 'Exited' instances in the validation set.

F1-Score for 'Exited': The F1-Score is 0.618, which is a balanced measure of the model's performance for the 'Exited' class, taking both precision and recall into account.

Model Performance Summary: Model_e ROC-AUC Tuning

The model is performing reasonably well in terms of accuracy and is particularly strong in correctly identifying those who have 'Not Exited.' However, with a precision of 0.549 and an F1-score of 0.618 for the 'Exited' class, there is still room for improvement. The False Positive rate is also relatively high at 11.88%, which might require attention. Given your focus on facts and precision, these are areas that could be fine-tuned for better performance.

Model Performance Summary:
Neural Network model_3 with Dropout

The architecture of model_3 is designed to be a Sequential neural network with multiple layers, aiming for better predictive performance. Here's the breakdown:
Input Layer: The model starts with an input layer that has 32 neurons and uses ReLU (Rectified Linear Unit) as the activation function. The input dimension is set to match the number of features in X_train.

Dropout Layer (1st): A dropout layer is added immediately after the first layer with a dropout rate of 0.2. Dropout layers are used to prevent overfitting by randomly setting a fraction of input units to 0 during training.

First Hidden Layer: The first hidden layer consists of 16 neurons and also uses ReLU as the activation function.

Model Performance Summary: Neural Network model_3 with Dropout

Dropout Layer (2nd): Another dropout layer is introduced, but this time with a dropout rate of 0.1.

Second Hidden Layer: This is the second hidden layer with 8 neurons, also using ReLU as the activation function.

Output Layer: Finally, the output layer has a single neuron with a Sigmoid activation function, which is suitable for binary classification problems like this one.

Model Performance Summary:
Neural Network model_3 with Dropout

Loss Function and Optimizer: The model will be compiled using the binary cross-entropy loss function and the Adam optimizer with a learning rate of 0.001 (not shown in the snippet, but assumed based on previous configurations).

This architecture is well-suited for a binary classification problem. It incorporates dropout to combat overfitting and employs ReLU activation functions for faster convergence during training. Overall, it's a balanced architecture that should offer a good compromise between learning capacity and generalization ability.

Model Performance Summary: Neural Network model_3 with Dropout

After defining the architecture of model_3, the next steps involve compiling and training the model.

Optimizer: The Adam optimizer is initialized with a learning rate of 0.001. Adam is a popular optimization algorithm that adapts learning rates during training, making it efficient and effective for a wide range of problems.

Compilation: The model is compiled using the binary cross-entropy loss function, which is the standard choice for binary classification tasks. The metric used for evaluation is accuracy.

Training: The model is trained on the training set X_train and y_train using a batch size of 32. The training will run for a maximum of 100 epochs, but it may stop early due to the Early Stopping callback.

Model Performance Summary:
Neural Network model_3 with Dropout

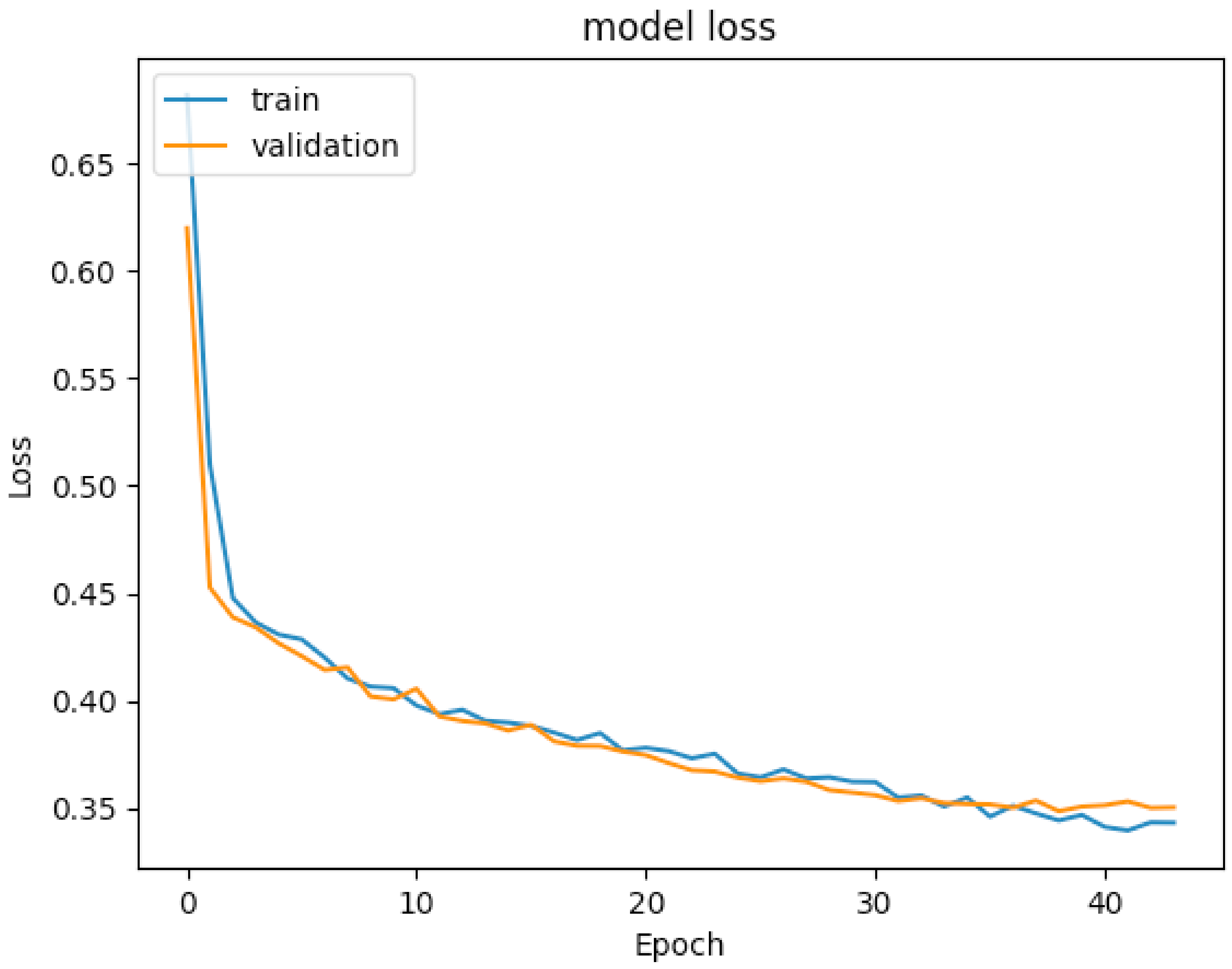
Validation Data: During training, the model's performance is also evaluated on a validation set (X_val, y_val) to monitor overfitting and adjust the model accordingly.

Callbacks: The Early Stopping callback (es_cb) is used to halt the training process if the model's performance on the validation set doesn't improve for a certain number of epochs (patience is set to 5). The monitored metric is validation loss (val_loss), and the training will stop if this doesn't improve by at least 0.001 (min_delta parameter).

Verbose: The verbosity level is set to 1, which means that progress bars and intermediate results will be displayed during training.

The combination of these elements aims to train a robust and accurate model while avoiding overfitting, thanks to the Early Stopping and Dropout layers. This should ideally result in a model that generalizes well to unseen data.

Model Performance Summary: Neural Network model_3 with Dropout



The training process terminated at 44 epochs, indicating that the early stopping mechanism found no substantial improvement in validation loss after a certain point.

The behavior of the loss and accuracy curves for both the training and validation sets being closely aligned is generally a good sign. It suggests that the model is neither underfitting nor overfitting.

Model Performance Summary: Neural Network model_3 with Dropout

Accuracy Curves: Similarly, the close alignment of training and validation accuracy indicates that the model performs consistently on both seen and unseen data.

Convergence: The fact that the lines for the training and validation sets crossed occasionally and continued together suggests that the model has reached a point of stability, further confirmed by the triggering of the early stopping mechanism.

The graphical trends indicate a well-fitted model. The early stopping at epoch 44 also suggests that additional training is unlikely to improve the model significantly, thus saving computational resources.

Model Performance Summary: Neural Network model_3 with Dropout

The best threshold of 0.241287 is the optimal cut-off point for maximizing both sensitivity and specificity. Any score above this threshold would be classified as 'Exited' and below as 'Not Exited'.

G-Mean: The score of 0.796 is quite strong, indicating a good balance between sensitivity and specificity. A G-Mean close to 1 suggests that the classifier performs well on both classes of the target variable.

The plot also shows a marker labeled 'Best' at the point where the largest G-Mean is found, providing a visual indication of the optimal model threshold.

This optimal threshold can now be used for making class predictions on new data.

Model Performance Summary:
Neural Network model_3 with Dropout

	Precision	Recall	F1-Score	Support
0	0.92	0.82	0.87	1274
1	0.51	0.73	0.6	326
Accuracy			0.8	1600
Macro avg	0.72	0.78	0.73	1600
Weighted avg	0.84	0.8	0.81	1600

True Negative (Class 0): The model has a high precision of 0.92, indicating that it is very accurate when predicting the negative class. The recall of 0.82 suggests that the model is able to identify a substantial majority of the actual negative instances.

True Positive (Class 1): The precision of 0.51 indicates a moderate level of false positives in the prediction. However, the recall of 0.73 is relatively high, which means the model is good at catching the positive instances.

Model Performance Summary: Neural Network model_3 with Dropout

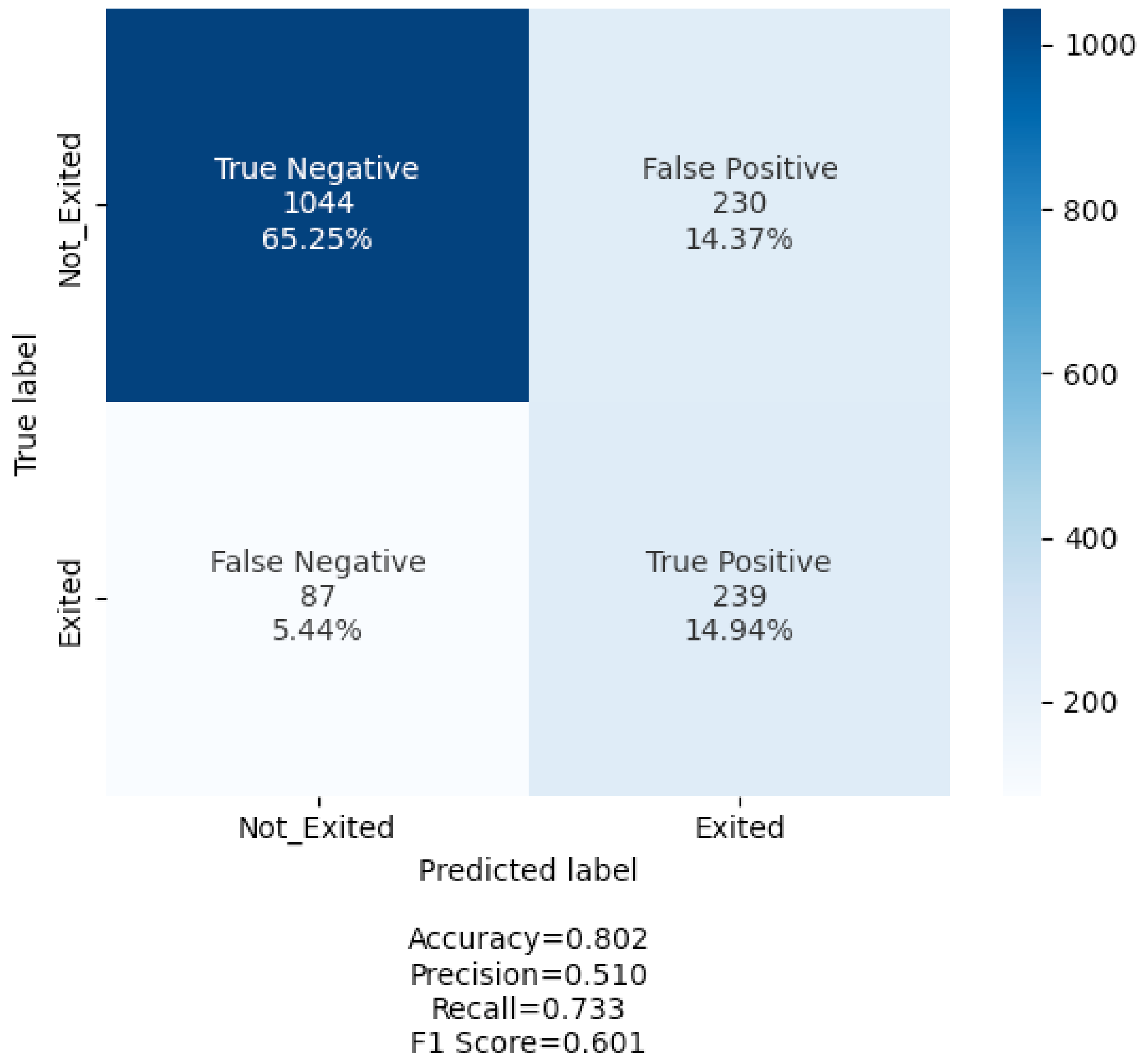
Accuracy: At 0.8, the overall accuracy is reasonably high, indicating that the model is correct 80% of the time across both classes.

Macro Average: The macro average for precision, recall, and F1-score are 0.72, 0.78, and 0.73 respectively. These values are close to each other, suggesting that the model is balanced and performs similarly across both classes.

Weighted Average: The weighted averages for precision, recall, and F1-score are 0.84, 0.8, and 0.81 respectively, which are consistent with the overall accuracy.

F1-Score: For Class 0 and Class 1, the F1-scores are 0.87 and 0.6 respectively. The F1-score combines both precision and recall into a single metric, and these scores indicate that the model is more reliable in predicting Class 0 (Not Exited) compared to Class 1 (Exited).

Model Performance Summary: Neural Network model_3 with Dropout



True Negatives: The model correctly identified 65.25% of the customers who did not churn.

This is a strong indicator of the model's capability to recognize loyal customers.

False Positives: At 14.37%, the rate of false positives is higher than ideal. While this isn't disastrous, it does indicate that the model occasionally misclassifies customers who will not churn as potential churn risks.

Model Performance Summary: Neural Network model_3 with Dropout

False Negatives: The model incorrectly classifies 5.44% of customers who actually did churn. This is a key metric, especially in a churn model, and it's relatively low, which is positive.

True Positives: The model correctly identifies 14.94% of customers who did churn. This is a crucial metric for a churn model, and the high value indicates strong performance in this area.

Accuracy: An overall accuracy of 80.2% is quite good and suggests that the model has generalized well from the training data to the unseen validation data.

Model Performance Summary:
Neural Network model_3 with Dropout

Precision: At 51%, the precision for identifying churn is moderate. This suggests that among the customers the model identifies as likely to churn, 51% actually do.

Recall: A recall of 73.3% is noteworthy. This metric is often considered more important than precision in churn models, as failing to identify customers who will churn can be more costly than false alarms. This is the highest recall we have observed so far among the different models.

F1 Score: At 0.61, the F1 Score is balanced, combining both precision and recall into a single measure of the model's accuracy.

Based on these metrics, this model has demonstrated the best performance so far in terms of recall. Given the cost implications of failing to identify a customer who will churn, this model would likely be the most useful for targeted customer retention efforts.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

Dropout: Dropout layers are included to prevent overfitting. The rate is set at 0.5 for the first dropout layer.

Learning Rate: The learning rate for the Adam optimizer can be tuned via the lr parameter.

Hidden Layers: Two hidden layers are defined, with the number of neurons in each layer specified by the layer_1 and layer_2 parameters.

Loss Function: Binary Cross-Entropy is used as this is a binary classification problem.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

GridSearchCV: This is employed to tune the model's hyperparameters, specifically the batch size and learning rate.

Grid Search Parameters:

Batch Size: Grid search will try 40, 64, and 128 as possible batch sizes.

Learning Rate: Three possible learning rates (0.01, 0.001, 0.1) will be tested.

Cross-Validation: The dataset will be split into 3 folds for cross-validation.

Number of Jobs: We've set n_jobs=-1 which means the computation will be dispatched on all CPUs to perform the tasks in parallel.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

By running a GridSearchCV, we are going to be able to find the best combination of batch size and learning rate for our model, based on the defined search grid. This should ideally improve the performance of your model.

Fitting the model 3 times for each of the 9 combinations of hyperparameters, totaling 27 fits.

According to the results, the best model achieved an accuracy of approximately 81.69% on the validation set. The best hyperparameters for this model were a batch size of 40 and a learning rate of 0.01.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

Key Takeaways:

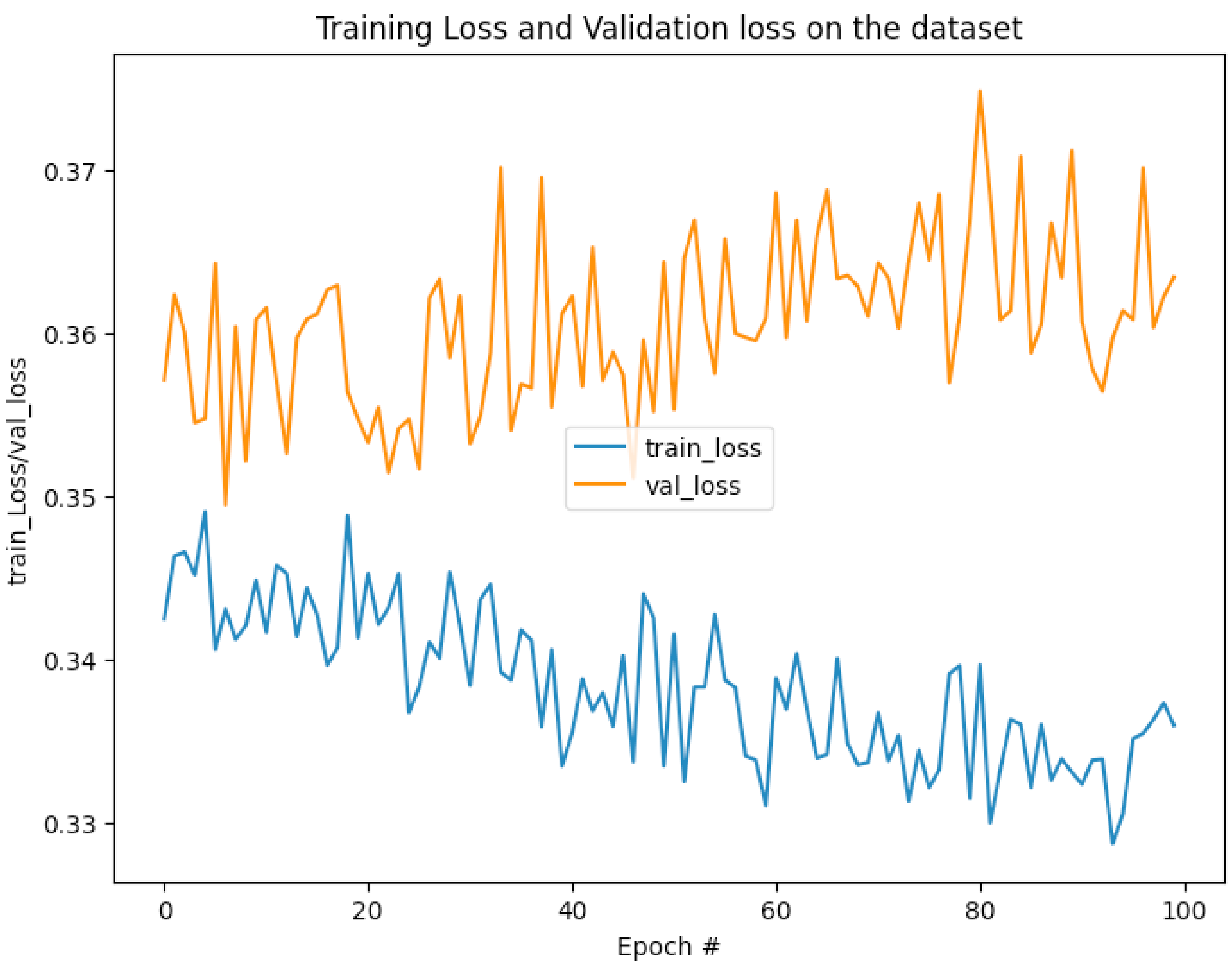
Best Batch Size: The optimal batch size, according to the grid search, is 40. A smaller batch size can offer a regularizing effect and lower generalization error.

Best Learning Rate: A learning rate of 0.01 was found to be the best. This is a critical hyperparameter as it controls the update step during training.

Validation Accuracy: The model achieved an accuracy of approximately 81.69% on the validation set, which indicates a good fit.

Loss and Accuracy: The model's loss on the training set was approximately 0.4558, and the accuracy was approximately 80.30%. For the validation set, the loss was approximately 0.4211, and the accuracy was approximately 81.81%.

Model Performance Summary: Neural Network model_v2 with Hyperparameter tuning



The graph depicting training and validation loss over 100 epochs reveals some noteworthy patterns:

Irregularity: Both the training and validation loss lines are highly irregular, showing several peaks. This could indicate that the model is having difficulty converging.

Divergence: As epochs increase, the lines for training and validation loss begin to diverge. This is a classic sign of the model overfitting on the training data.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

With the same hyperparameters in Batch size and learning rate of 40 and 0.01 respectively, we were able to find the threshold of 0.227791 and a G-mean of 0.812 by running a ROC Curve

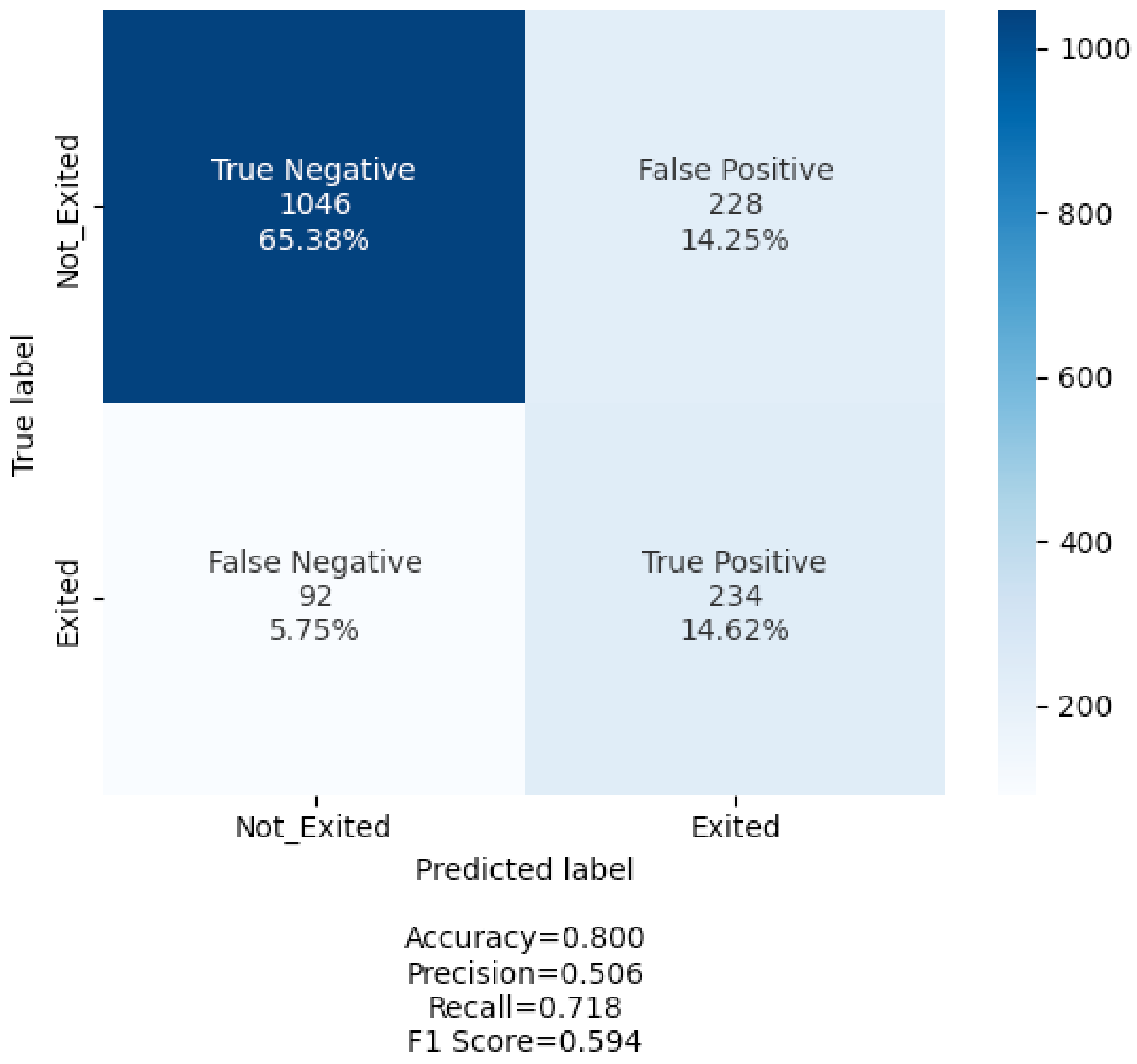
Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning

	Precision	Recall	F1-Score	Support
0	0.92	0.82	0.87	1274
1	0.51	0.72	0.6	326
Accuracy			0.8	1600
Macro avg	0.72	0.78	0.73	1600
Weighted avg	0.84	0.8	0.81	1600

- Accuracy: 0.8
- Precision for class 1: 0.51
- Recall for class 1: 0.72
- F1-Score for class 1: 0.6

The model demonstrates a decent accuracy of 80%. The recall of 72% for class 1 is particularly noteworthy, given that it's a key metric in this use-case.

Model Performance Summary:
Neural Network model_v2 with Hyperparameter tuning



Confusion Matrix:

- True Negatives: 1046 (65.38%)
- False Positives: 228 (14.25%)
- False Negatives: 92 (5.75%)
- True Positives: 234 (14.62%)

The model has a higher number of False Positives, which may be acceptable depending on the business context. The False Negatives, which are more concerning, are relatively low.

Recall is lower than previous model_3

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning

This model is constructed after balancing the dataset using the Synthetic Minority Over-sampling Technique (SMOTE). Balancing the classes is particularly important for improving the model's performance on the minority class, which, in this business context, is the "Exited" customers.

Model Components:

Input layer: 32 neurons, ReLU activation function.
Dropout layer: Rate of 0.2.
Hidden Layer 1: 16 neurons, ReLU activation function.
Dropout layer: Rate of 0.1.
Hidden Layer 2: 8 neurons, ReLU activation function.
Output Layer: 1 neuron, Sigmoid activation function.

Model Performance Summary: Neural Network model_v4 SMOTE & Hyperparameter tuning

By adding dropout layers with rates of 0.2 and 0.1, we introduce a form of regularization to the model. This helps in preventing overfitting, especially when we are dealing with a balanced dataset post-SMOTE.

Configuration:

Activation function for the final layer: Sigmoid, suitable for binary classification.

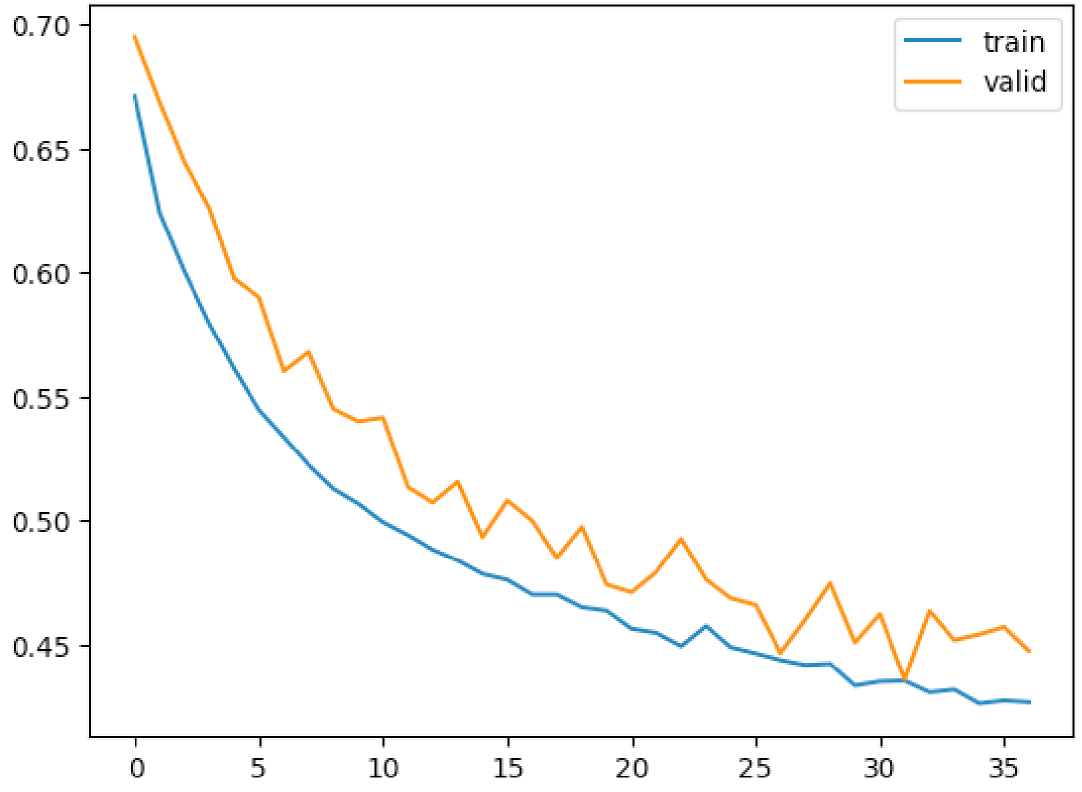
SMOTE is often used to improve the model's ability to generalize well to new, unseen data, especially for the minority class. Given that we're dealing with a business case where identifying the minority class (customers likely to exit) is crucial, using a balanced dataset could significantly improve the model's performance in this aspect.

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning

Early stopping is applied to monitor val_loss with a min_delta of 0.001 and patience of 5. This helps to halt the training process if the model starts overfitting.

Optimizer: Adam with a learning rate of 0.001.
Loss Function: Binary Cross-Entropy.
Metrics: Accuracy.
Batch Size: 64.
Epochs: 100.

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning



The plot indicates that the training and validation loss go parallel for the most part, showing that the model generalizes well on the validation set. The 'peaky' behavior might suggest some variability in the learning process, but it's not diverging, which is a good sign.

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning

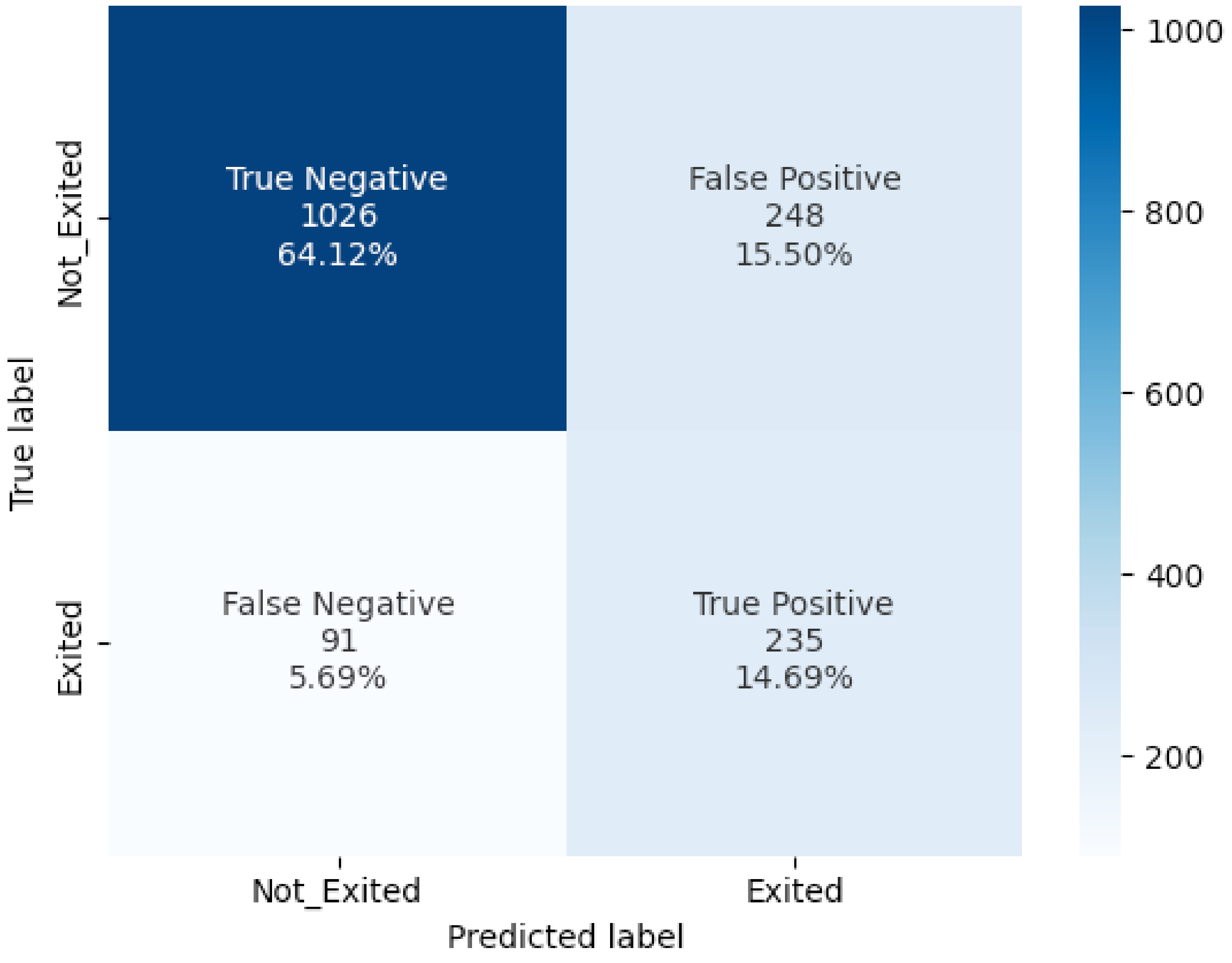
ROC Curve Analysis:
Best Threshold for classification: 0.472441.
Geometric Mean (G-Mean): 0.830.
The high G-Mean score indicates that the model performs well on both the minority and majority classes. The threshold is higher in this case so any prediction over 0.472441 is going to be classified as churn.

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning

	Precision	Recall	F1-Score	Support
0	0.92	0.81	0.86	1274
1	0.49	0.72	0.58	326
Accuracy			0.79	1600
Macro avg	0.7	0.76	0.72	1600
Weighted avg	0.83	0.79	0.8	1600

Recall of 0.72: This is significant because recall is our primary metric of interest. It suggests that the model is capable of identifying 72% of the total actual positives, which is crucial for a customer churn model.
Precision of 0.49: This indicates that among the instances predicted as positive, 49% are actually positive. While not as high as we might like, it is a trade-off we make for a higher recall.
F1-Score of 0.58: The F1-Score is a good balance between precision and recall, and a score of 0.58 indicates that the model is reasonably well-balanced.
Accuracy of 0.79: Though accuracy is not our primary metric, a score of 0.79 is a reasonable indicator of the overall performance of the model.

Model Performance Summary:
Neural Network model_v4 SMOTE & Hyperparameter tuning



Accuracy=0.788
Precision=0.487
Recall=0.721
F1 Score=0.581

This model could be quite effective for early identification of customers who are likely to churn. The high recall rate is particularly valuable for this application, as it enables proactive customer retention initiatives. While the precision is not particularly high, in a churn model, false positives may be less of a concern than false negatives, as the cost of false negatives (losing a customer) can be much higher than the cost of false positives (unnecessary retention efforts). Therefore, from a business standpoint, this model appears to be quite useful.

Model Performance Summary: Final Model

We are choosing model_3 as our best model as we observed the best recall score.

Wea re going to find the ROC Curve to apply it to model_3

ROC Curve and G-Mean:

- Best Threshold: 0.242378
- G-Mean: 0.808

The ROC curve and the G-Mean value indicate that the model performs significantly well in distinguishing between the positive and negative classes. A G-Mean score of 0.808 suggests a well-balanced model in terms of sensitivity and specificity.

Model Performance Summary: Final Model

	Precision	Recall	F1-Score	Support
0	0.93	0.81	0.86	1593
1	0.5	0.74	0.6	407
Accuracy			0.8	2000
Macro avg	0.71	0.78	0.73	2000
Weighted avg	0.84	0.8	0.81	2000

Model Performance Summary: Final Model

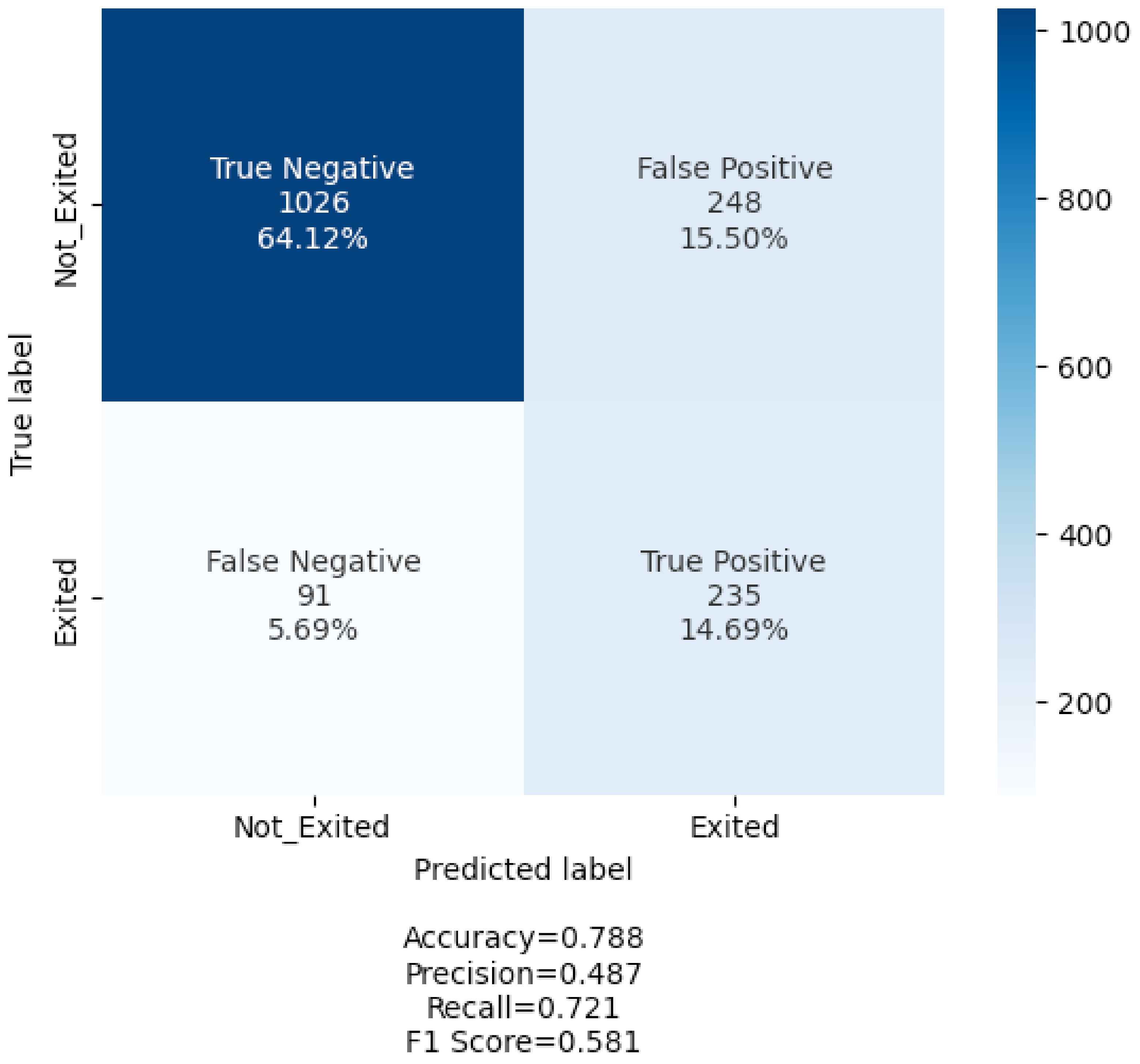
High Recall of 0.74: This model excels in identifying the customers who are actually going to churn. A 74% recall rate is excellent for proactive customer retention strategies.

Moderate Precision of 0.5: This metric indicates that 50% of the identified churn cases will indeed churn. While precision could be improved, the current level may be acceptable given the high recall rate, depending on the business context.

F1-Score of 0.6: The F1-Score is a balanced metric and a value of 0.6 indicates that the model is relatively well-balanced between precision and recall.

Accuracy of 0.8: Although accuracy is not our primary concern for this imbalanced problem, an 80% accuracy rate is still an excellent indicator of the model's general capability to classify instances correctly.

Model Performance Summary: Final Model



Model_3 stands out as the most well-balanced and effective for the task at hand, particularly for its high recall rate. From a business perspective, this model would be highly beneficial for identifying potential churn and taking preventive measures.

APPENDIX

Data Background and Contents

The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Data Dictionary

CustomerId: Unique ID which is assigned to each customer
Surname: Last name of the customer
CreditScore: It defines the credit history of the customer.
Geography: A customer’s location
Gender: It defines the Gender of the customer
Age: Age of the customer
Tenure: Number of years for which the customer has been with the bank
NumOfProducts: It refers to the number of products that a customer has purchased through the bank.
Balance: Account balance
HasCrCard: It is a categorical variable that decides whether the customer has a credit card or not.

Data Background and Contents

EstimatedSalary: Estimated salary

isActiveMember: It is a categorical variable that decides whether the customer is an active member of the bank or not (Active member in the sense, using bank products regularly, making transactions, etc)

Exited: It is a categorical variable that decides whether the customer left the bank within six months or not. It can take two values

0=No (Customer did not leave the bank)

1=Yes (Customer left the bank)

Great Learning

Happy Learning !