INN Project

Neural Networks – Bank Churn Project

December 15, 2024

Contents / Agenda

Business Problem Overview

- Background and Context:

  Businesses like banks that provide service have to worry about the problem of 'Churn' i.e. customers leaving and joining another service provider. It is important to understand which aspects of the service influence a customer's decision in this regard. Management can concentrate efforts on the improvement of service, keeping in mind these priorities.

- Objective:
  - Given any Bank customer, we must build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.

Solution Approach (Step 1 – Data Exploration)

- **Objective**: Understand the dataset, identify patterns, and handle missing or inconsistent data.
- **Actions**:
  - Load and preview the dataset.
  - Check for missing or null values.
  - Understand the distribution of numerical and categorical variables.
  - Evaluate correlations among features.
  - Visualize the distribution of the target variable (Exited).

Solution Approach (Step 2 – Data Preprocessing)

Objective: Prepare the dataset for machine learning by cleaning and encoding.

- **Drop irrelevant features**: CustomerId and Surname do not contribute to the prediction task.
- **Encode categorical variables**:
    - Convert Gender and Geography into numerical formats (e.g., one-hot encoding for Geography, binary encoding for Gender).
- **Normalize numerical features**:
    - Scale features like CreditScore, Age, Balance, and EstimatedSalary to ensure the model trains effectively.
- **Handle imbalances**:
    - Check for class imbalance in the Exited column. If imbalanced, use techniques like oversampling, undersampling, or SMOTE.
- **Split data**:
    - Divide the dataset into training, validation, and test sets. (80% Training, 20% Test per usual)

Solution Approach (Step 3 – Build the NN Model)

- **Objective**: Create a robust classifier to predict churn.
- Define a neural network architecture:
    - Input layer matching the number of features.
    - Hidden layers with activation functions (e.g., ReLU).
    - Dropout layers to prevent overfitting.
    - Output layer with a single neuron and a sigmoid activation function (binary classification).
- Compile the model:
    - Loss function: Binary cross-entropy.
    - Optimizer: Adam optimizer.
    - Metrics: Accuracy, Precision, Recall, or F1-Score.

Solution Approach (Step 4 – Model Performance Evaluation)

- **Objective**: Optimize the model using the training and validation sets.
  - Train the model using mini-batch gradient descent.
  - Use early stopping to prevent overfitting by monitoring validation loss.
  - Evaluate the model on the test set using metrics like accuracy, confusion matrix, and AUC-ROC
  - We will evaluate the model primarily on Recall and secondly on F-1 Values:
    - Using **recall** as the evaluation metric is a great choice for this business scenario because it emphasizes identifying churned customers (true positives) and minimizes false negatives. In churn prediction, missing a potential churner can be costly for the business.
    - **Recall** ensures churners (minority class) are identified (higher TP, lower FN)
    - **Precision**: Slightly reduced but still balanced due to regularization (Dropout).

Solution Approach (Step 4 – Model Performance Evaluation)

- **Performance Comparison and Reasoning**
- For each model, we will collect the following:
  - **Training vs Validation Loss**: Assess overfitting.
  - **Recall, Precision, and F1-score**: To gauge the ability to predict churn.
  - **AUC-ROC Curve**: For threshold-independent performance.
    - Compare performance to select the best model.
- **Choose the Best Model**
    - Reason for selection could be:
      - Best recall on validation data (important for churn prediction).
      - Balanced precision-recall trade-off.
      - Generalization ability (minimal overfitting).

Solution Approach (Step 4 – Model Performance Evaluation)

Performance Expectations by Model

1. **Model 1 (Adam optimizer)**:

   Likely to perform better than a basic SGD model but could struggle with class imbalance, leading to lower recall for churners.
2. **Model 2 (Adam optimizer + Dropout)**:

   Improved generalization due to Dropout but still affected by the imbalance, limiting recall for churners.
3. **Model 3 (SMOTE + SGD optimizer)**:

   Balancing via SMOTE improves recall but might converge slower and less effectively due to SGD's limitations in complex optimization.
4. **Model 4 (SMOTE + Adam optimizer)**:

   Likely to outperform Model 3, as Adam is more efficient and effective with imbalanced data.
5. **Model 5 (SMOTE + Adam optimizer + Dropout)**:

   Combines the strengths of SMOTE, Adam optimizer, and Dropout for optimal recall and F1 performance.

Executive Summary (Business Insights)

1. **Customer Segmentation**:
    - **Demographic Features**: Variables like Age, Gender, and Geography impact churn differently. For example, younger customers might churn more due to competitive offers, while older customers may value stability.
    - **Behavioral Features**: Variables like Balance, Tenure, and NumOfProducts give insights into customer engagement. Customers with low tenure or fewer products are more likely to churn.
2. **Credit and Financial Health**:
    - **Credit Score**: A low credit score might correlate with a higher likelihood of churn due to financial difficulties or dissatisfaction.
    - **Balance and Estimated Salary**: Customers with high balances and low engagement could be at risk of seeking better returns or services elsewhere.
3. **Customer Engagement**:
    1. **Activity Level**: The IsActiveMember feature highlights engagement with bank services. Inactive members are more prone to churn

Executive Summary (Business Insights)

3. **Customer Engagement**:

    - **Activity Level**: The IsActiveMember feature highlights engagement with bank services. Inactive members are more prone to churn.

    - **HasCrCard**: Having a credit card might indicate deeper integration with the bank, reducing churn likelihood.

4. **Geography**:

    - Churn rates often vary across regions due to competition, local economic conditions, or customer satisfaction levels with branch services.

Executive Summary (Recommendations)

- **1. Focus on Retention Efforts for At-Risk Customers**
    - Use churn predictions to segment customers into risk levels.
- For high-risk customers:
    - Offer personalized retention strategies (e.g., discounts, rewards).
    - Provide dedicated account managers to improve satisfaction.
- For medium-risk customers:
    - Increase communication about benefits and services.
    - Offer product bundling to encourage deeper engagement.

Executive Summary (Recommendations)

- **1. Focus on Retention Efforts for At-Risk Customers**
- Predicting churn allows the bank to segment customers by their likelihood of leaving. Retention is more cost-effective than acquiring new customers, making this a key business strategy.
- **Proactive Outreach**:
  - Identify high-risk customers using the churn prediction model and reach out with retention offers.
  - For example, a customer with low tenure, minimal balance, or no credit card might receive a personalized call or email offering a promotional rate or financial consultation.
- **Custom Rewards**:
  - Offer tailored loyalty rewards, such as waiving fees or increasing interest rates for customers with high balances.
- **Improve Customer Service**
  - Assign relationship managers to high-value customers who are at risk.

Executive Summary (Recommendations)

- **2. Improve Engagement with Bank Products**
  - Promote products like credit cards or additional accounts to increase product penetration (captured by NumOfProducts).
  - Features like NumOfProducts and IsActiveMember highlight customer engagement. The more products a customer uses, the deeper their integration with the bank, reducing churn likelihood.
- **Cross-Sell Products**:
  - For customers with only one product, offer bundles or promotions to encourage them to use additional services, such as credit cards, investment accounts, or loans.
- **Educational Campaigns**:
  - Educate customers on the benefits of being active members, such as loyalty points or access to premium features.
- **Gamify Engagement**:
  - Introduce gamified reward programs where customers earn points for using services frequently

Executive Summary (Recommendations)

- **3. Address Customer Pain Points**
- Dissatisfaction often leads to churn. Identifying the specific pain points, whether financial, service-related, or competitive, is critical.
- **Recommendations:**
- **Analyze Churn Reasons**:
  - Use surveys or direct feedback to understand why customers leave.
  - Segment this data by geography, demographics, and product usage.
- **Provide Financial Support**:
  - For customers with low credit scores, offer credit-building products or free financial consultations to reduce frustration.
- **Localized Improvements**:
  - If churn is geographically skewed, invest in improving services in those regions (e.g., faster transaction times, better branch support).

Executive Summary (Recommendations)

- **4. Leverage Financial Data**
- High-value customers with significant balances are crucial to retain. However, they might churn if they perceive better opportunities elsewhere.
- **Premium Services**:
  - Offer high-value customers premium accounts with exclusive benefits (e.g., lower fees, higher savings interest rates, or personalized financial advice).
- **Interest Rate Adjustments**:
  - Provide competitive interest rates for customers with significant balances.
- **Tailored Financial Solutions**:
  - Recommend tailored investment or wealth management products to customers with substantial savings.

Executive Summary (Recommendations)

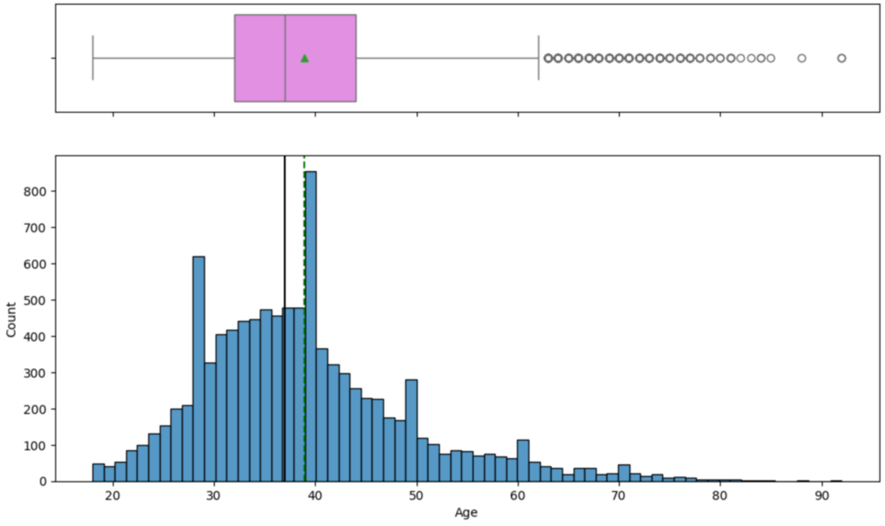
- **5. Evaluate Offers from Competitors**
- Competitor banks may offer better interest rates, fee structures, or benefits, enticing customers to switch.
- **Market Research**:
  - Continuously monitor competitor offerings and adjust the bank's rates and benefits accordingly.
- **Price Match Campaigns**:
  - Introduce "price matching" campaigns where the bank matches competitor offers for high-risk customers.
- **Customer Awareness**:
  - Highlight the unique benefits of staying with the bank (e.g., customer service quality, account security, or bundled services).

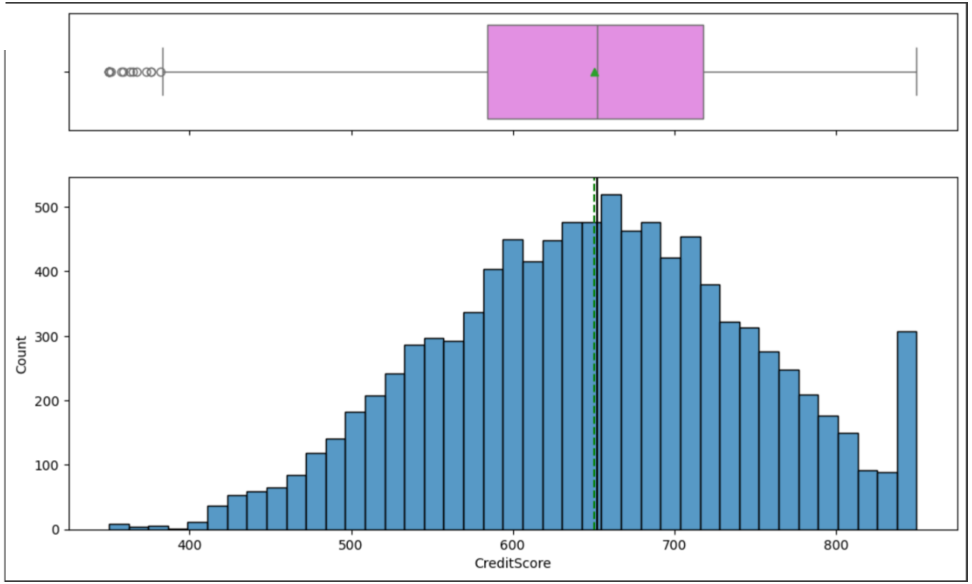
Executive Summary (Recommendations)

- **6. Key Metrics for Success**
- To measure the effectiveness of these strategies, track:
    - **Customer Retention Rate**: Percentage of customers retained over a period.
    - **Customer Lifetime Value (CLV)**: Increase in long-term revenue from retained customers.
    - **Churn Reduction**: Year-over-year decline in churn rate.
    - **Engagement Metrics**: Growth in active members and average products per customer.

EDA Results – Univariate Analysis

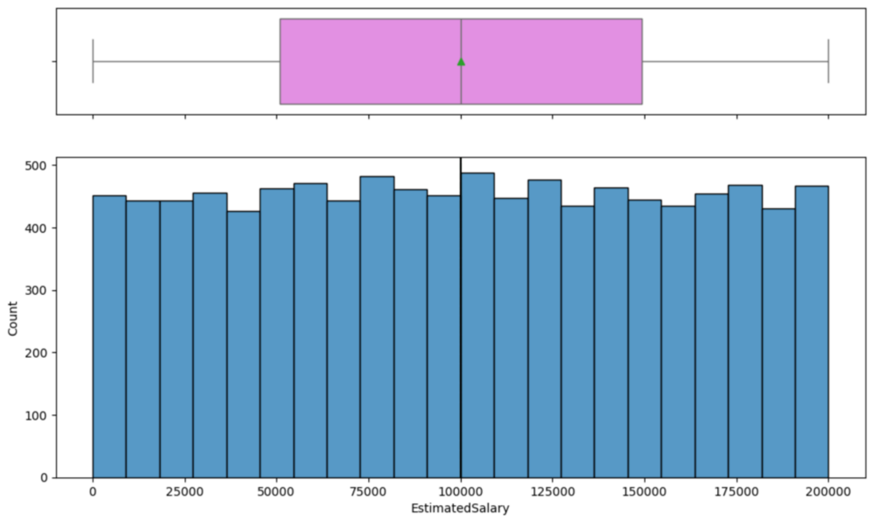
Age and Credit Score Boxplots from raw data

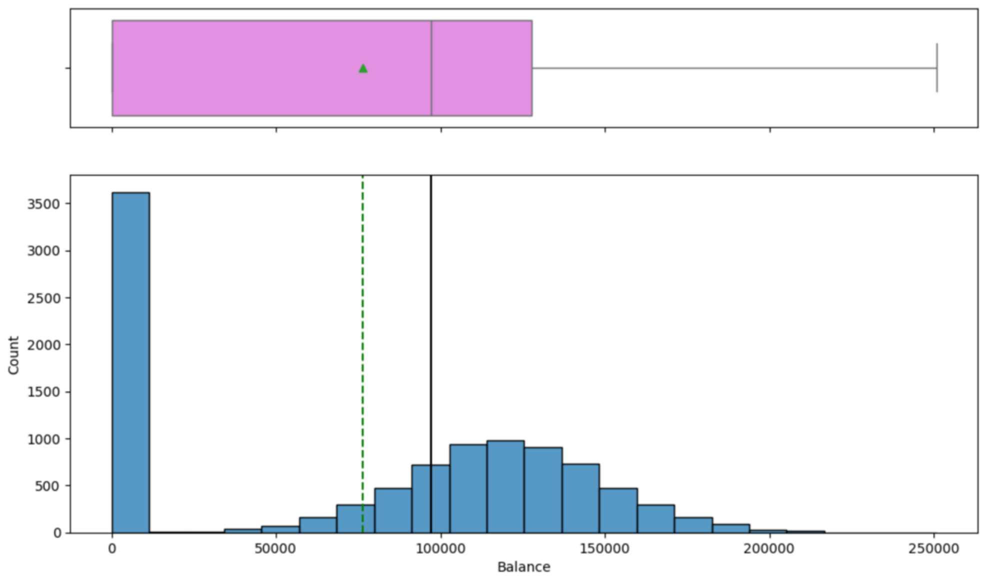




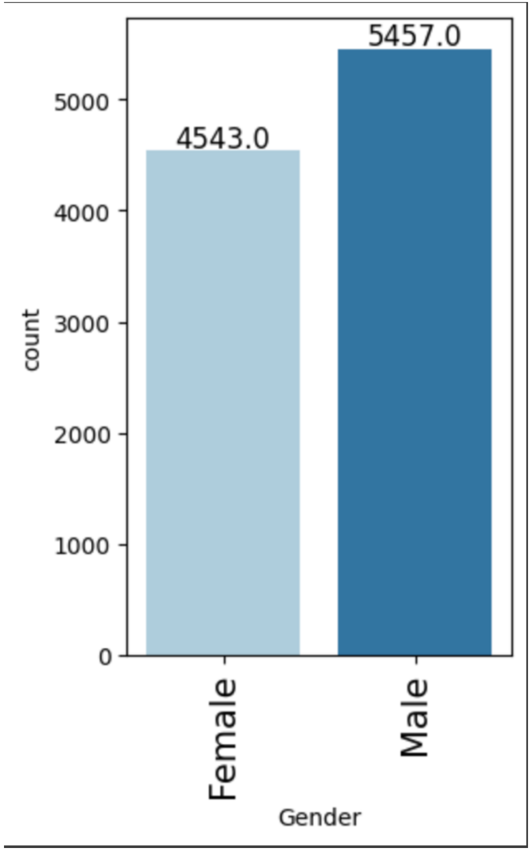
EDA Results – Univariate Analysis

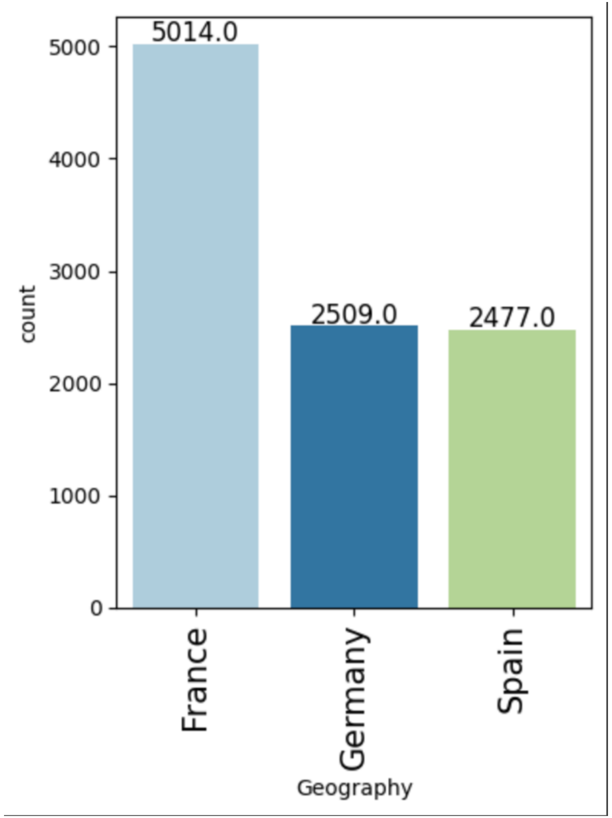
Estimated Salary and Credit Card Balance Box Plots

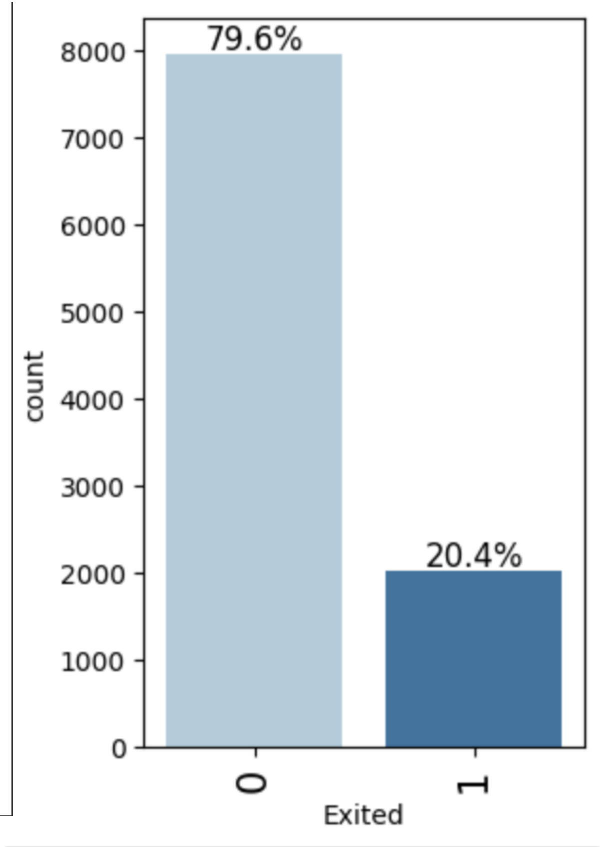




EDA Results – Univariate Analysis

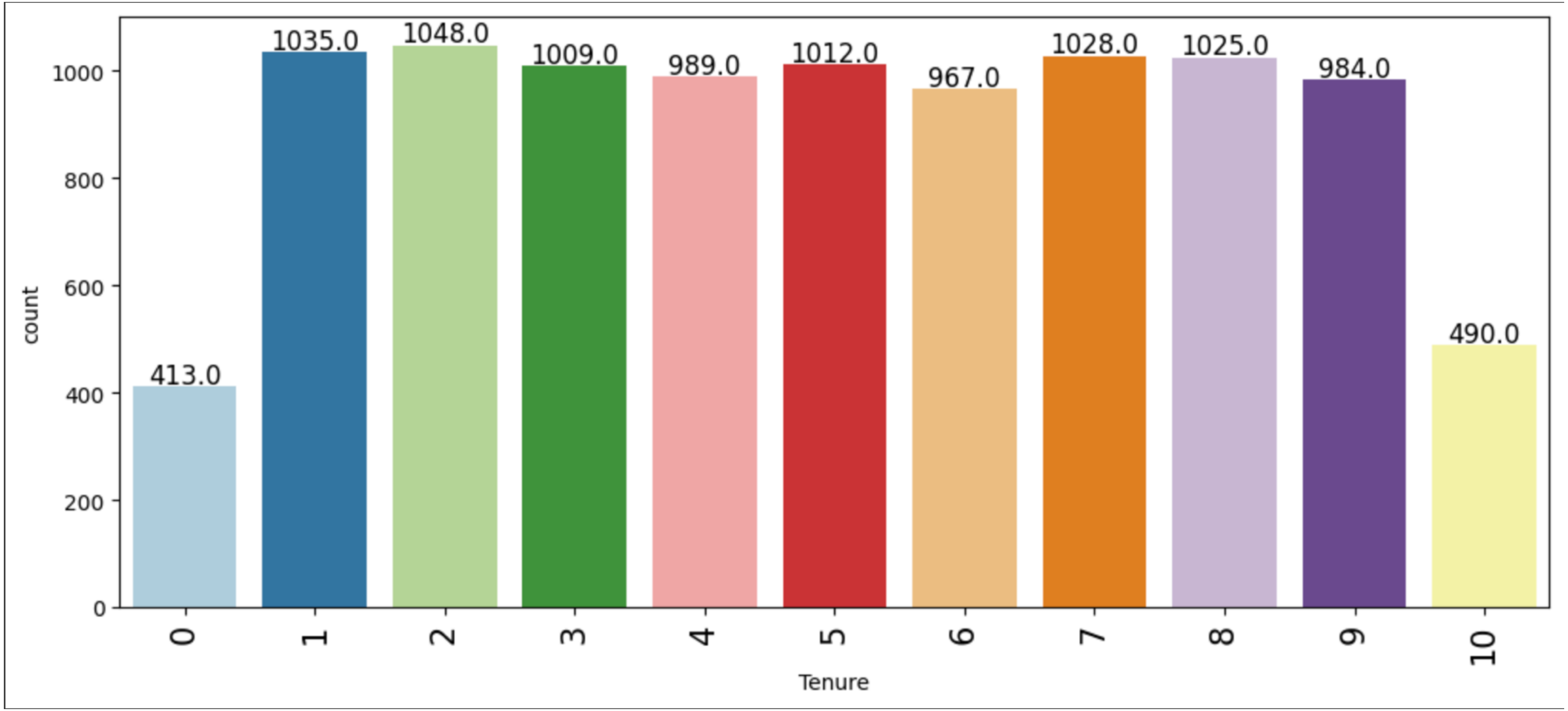






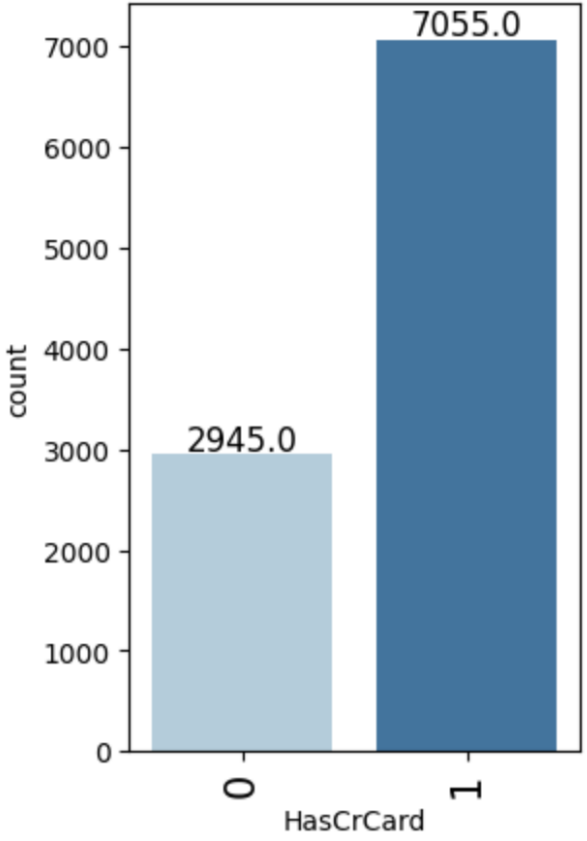
EDA Results – Univariate Analysis

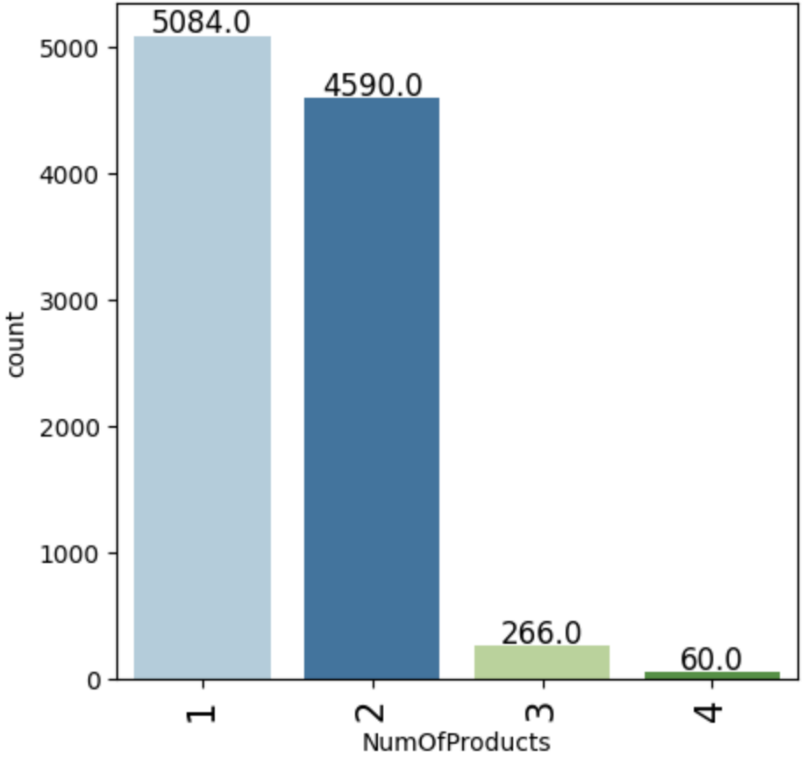
Graph showing how many customers exit after a given number of years.



EDA Results – Univariate Analysis

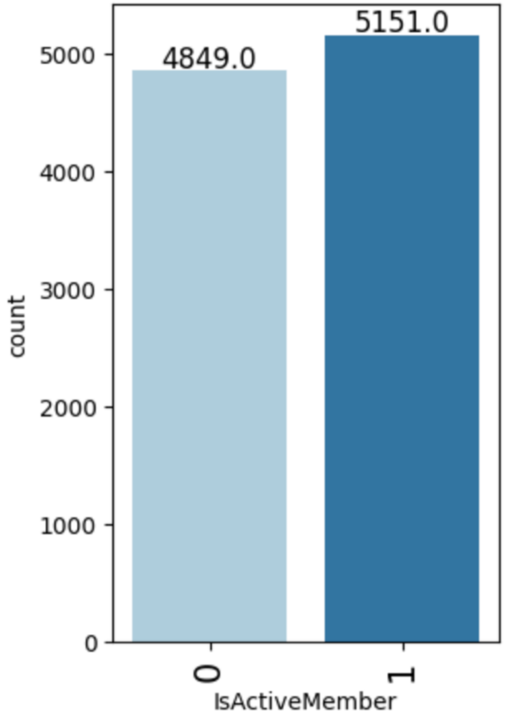
Count of those who have credit cards and number of products

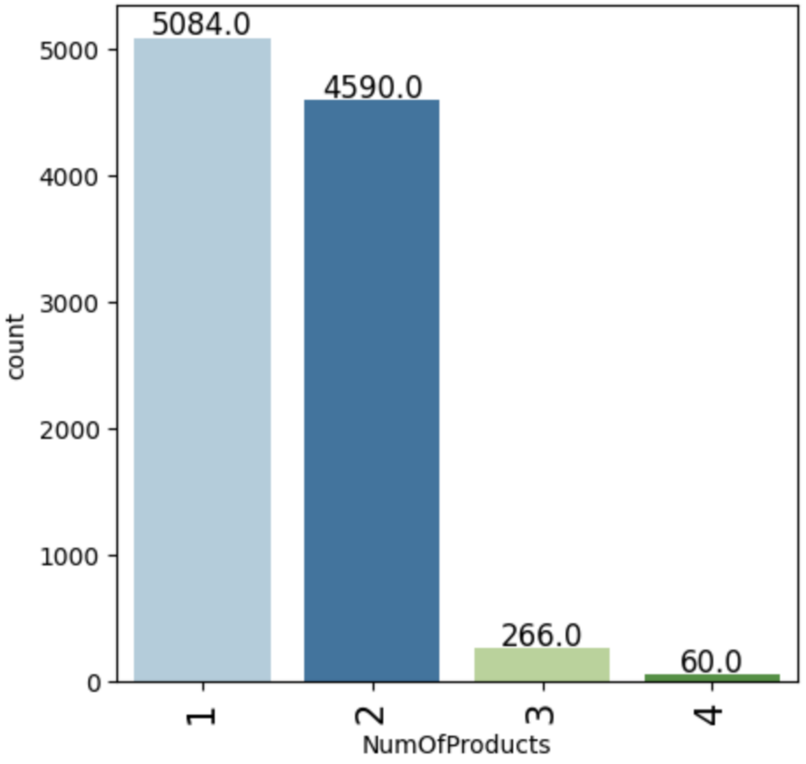




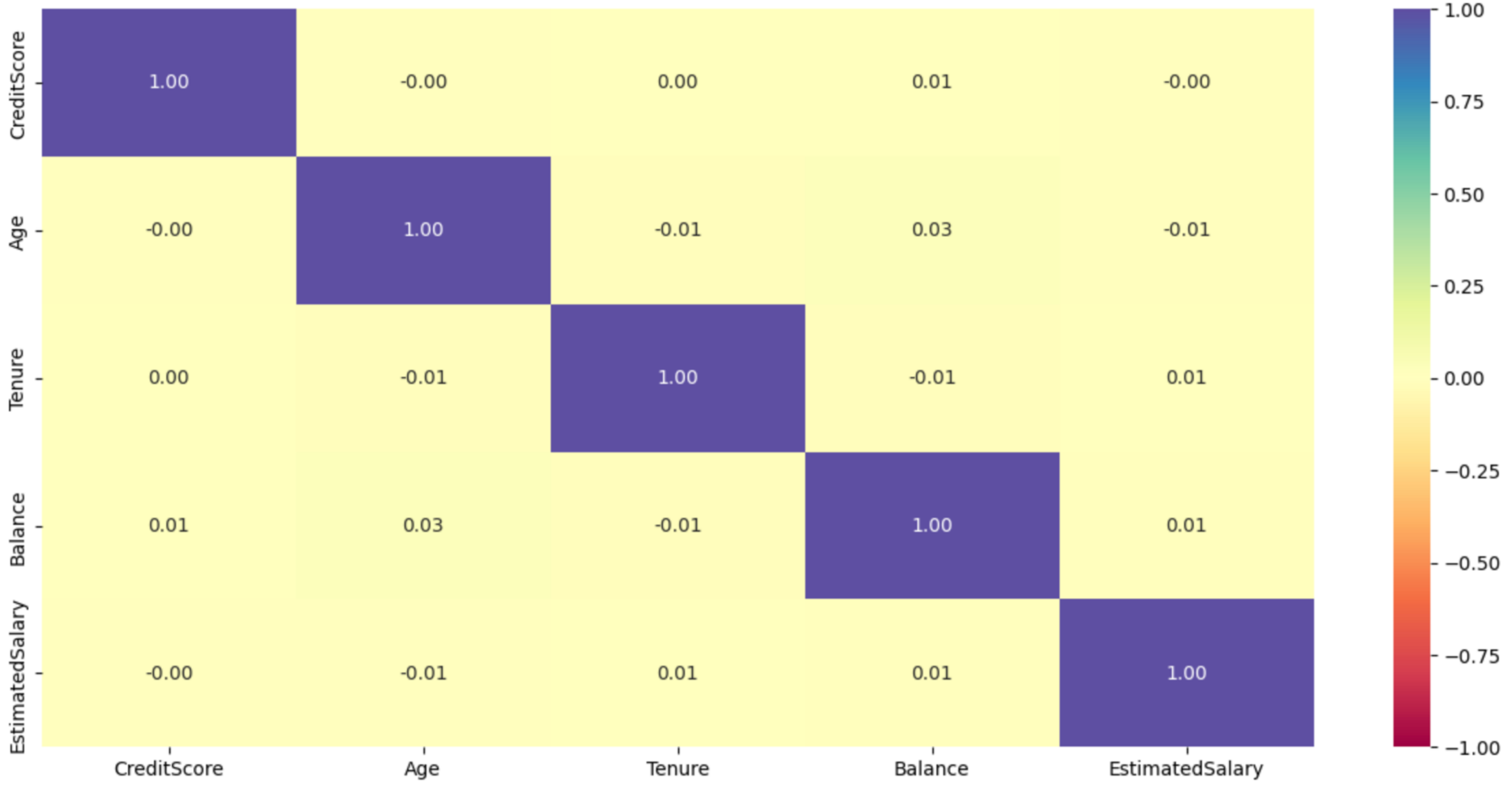
EDA Results – Univariate Analysis

Count for Active Members and Number of Products



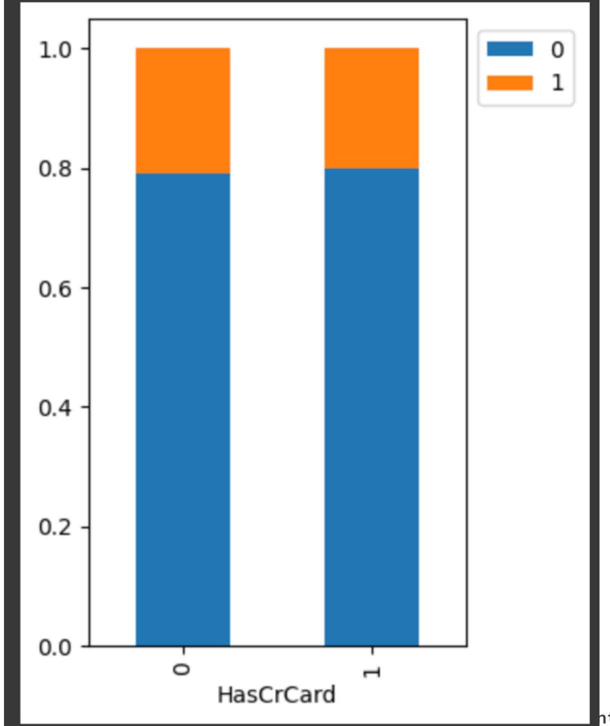


EDA Results – Bivariate Analysis (Heatmap)

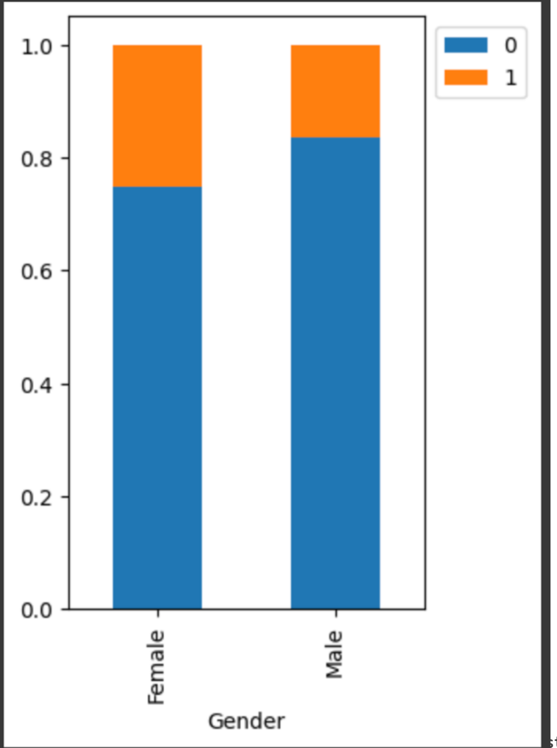


EDA Results – Bivariate Analysis

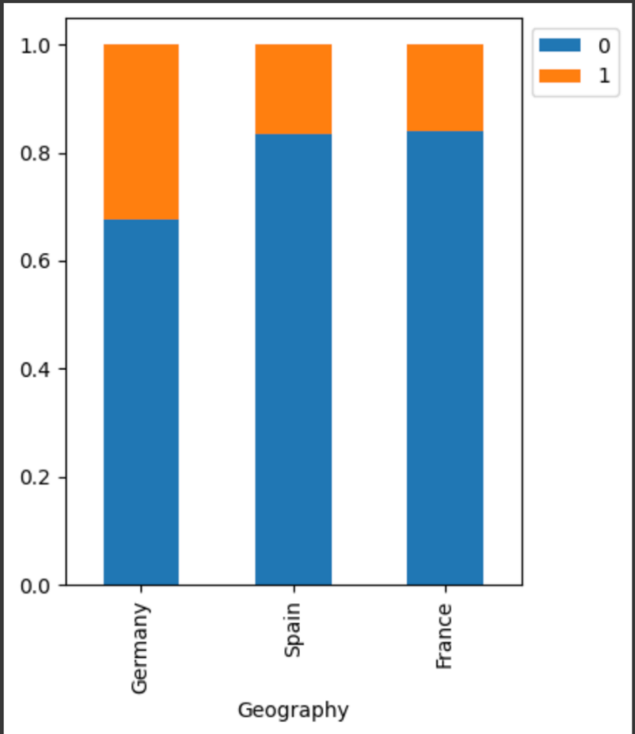
Exited	0	1	All
HasCrCard			
All	7963	2037	10000
1	5631	1424	7055
0	2332	613	2945



Exited	0	1	All
Gender			
All	7963	2037	10000
Female	3404	1139	4543
Male	4559	898	5457



Exited	0	1	All
Geography			
All	7963	2037	10000
Germany	1695	814	2509
France	4204	810	5014
Spain	2064	413	2477



EDA Results – Bivariate Analysis

The boxplots for numerical features reveal potential outliers in columns like:

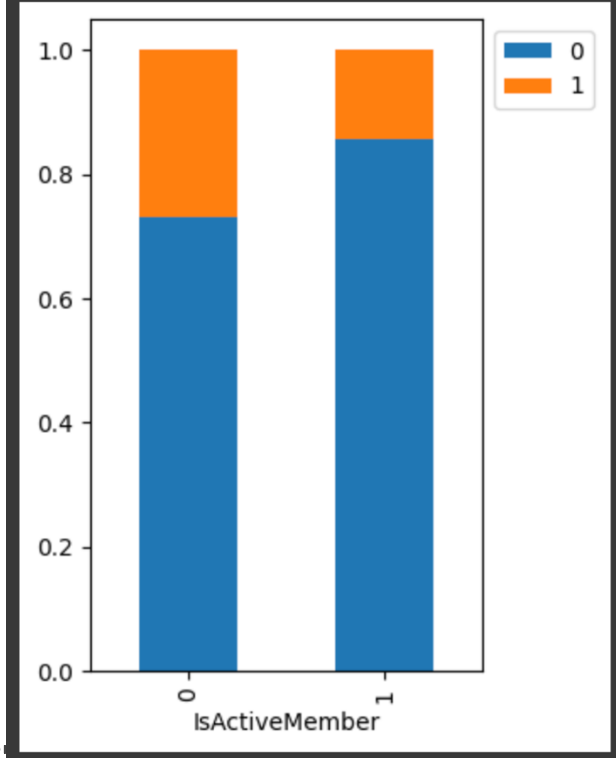
•**Balance**: Values near zero might indicate inactive accounts or unusual customer activity.

•**EstimatedSalary**: Outliers at the high end, though it seems to have a broader range by nature.

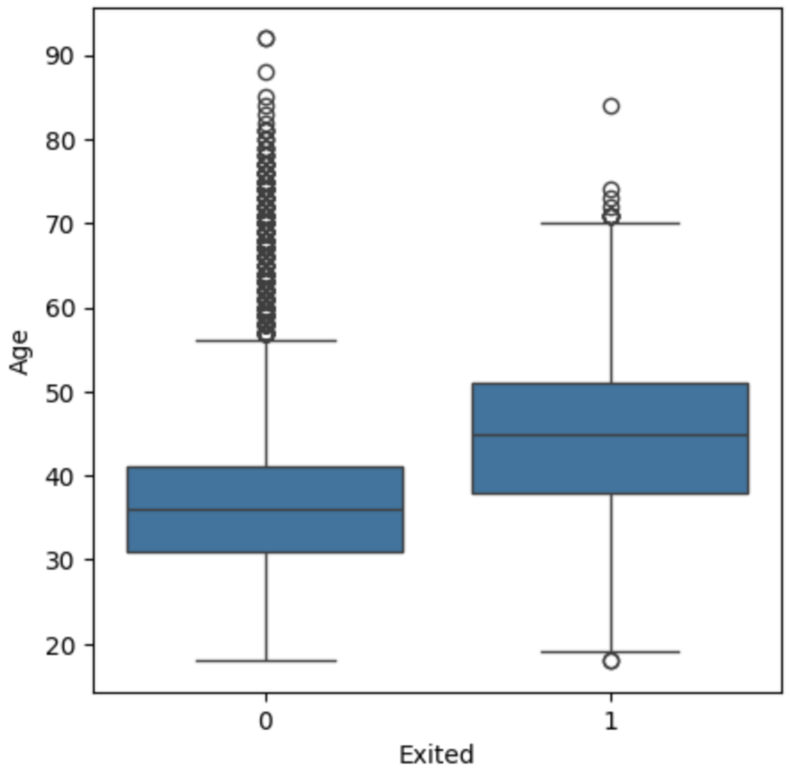
•**Age**: Possible outliers for extremely young or very old customers.

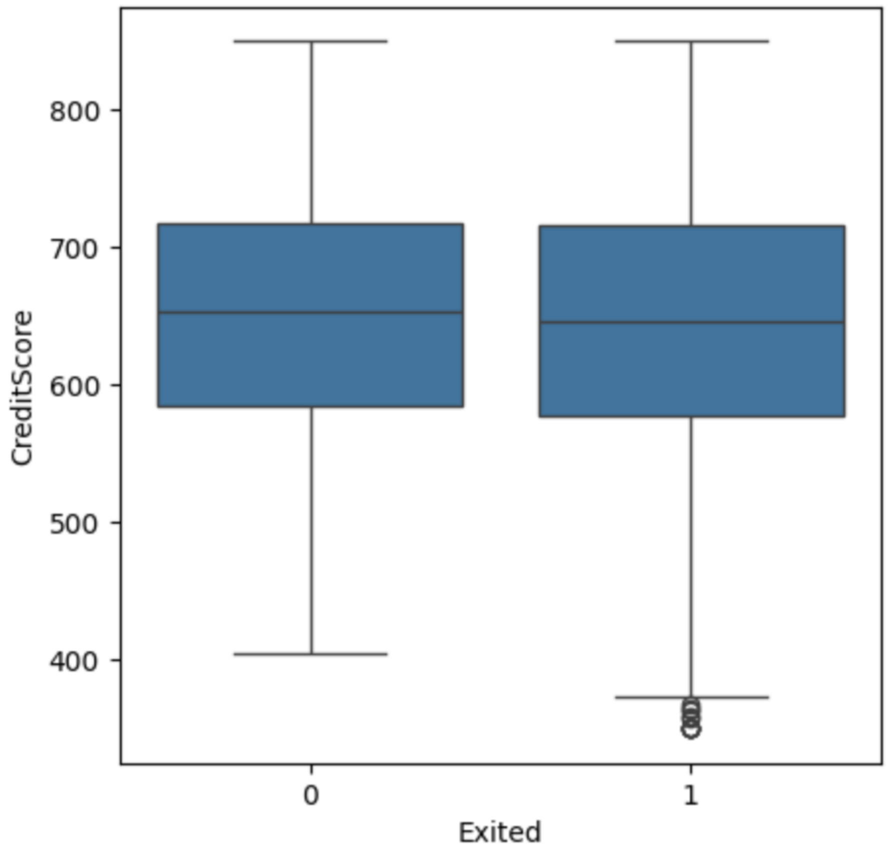
•**CreditScore**: Few potential outliers below the lower end of the range.

Exited	0	1	All
IsActiveMember			
All	7963	2037	10000
0	3547	1302	4849
1	4416	735	5151

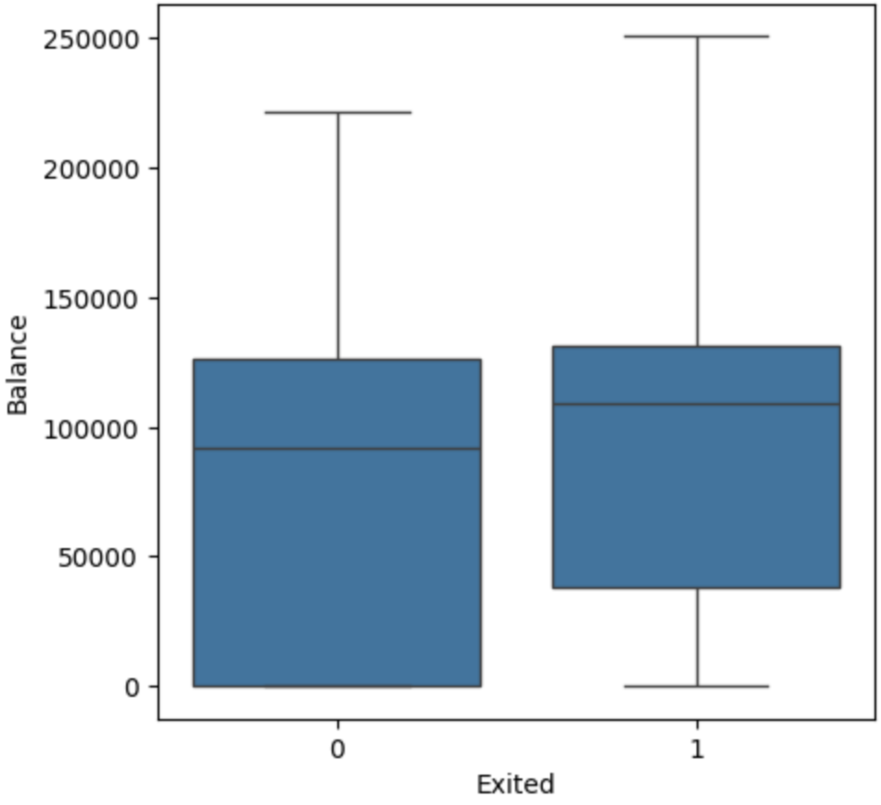


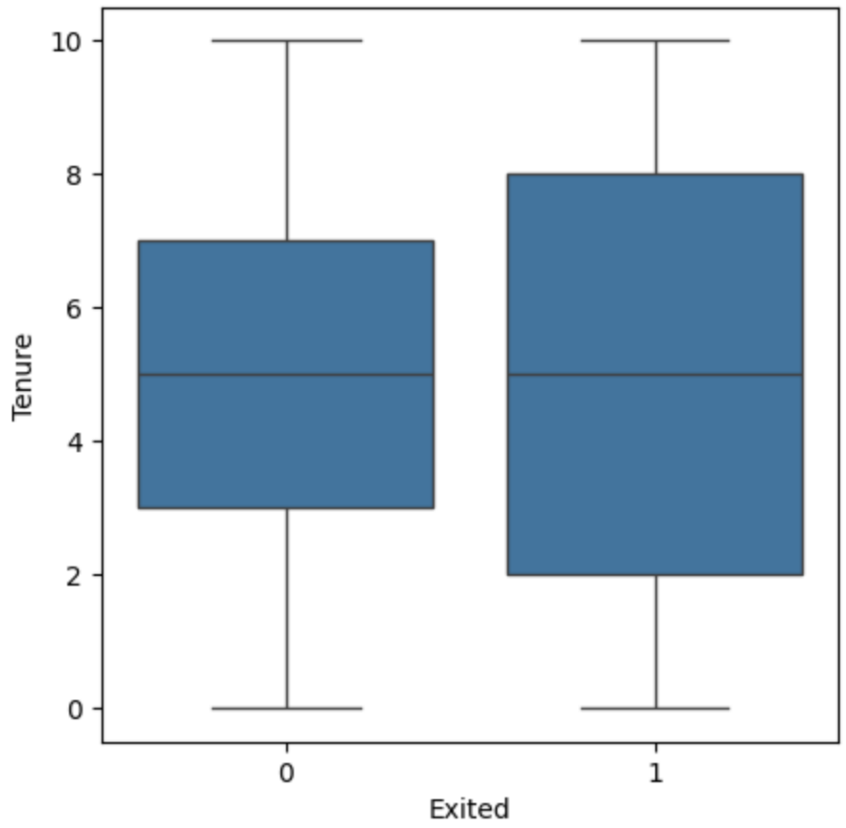
EDA Results – Bivariate Analysis



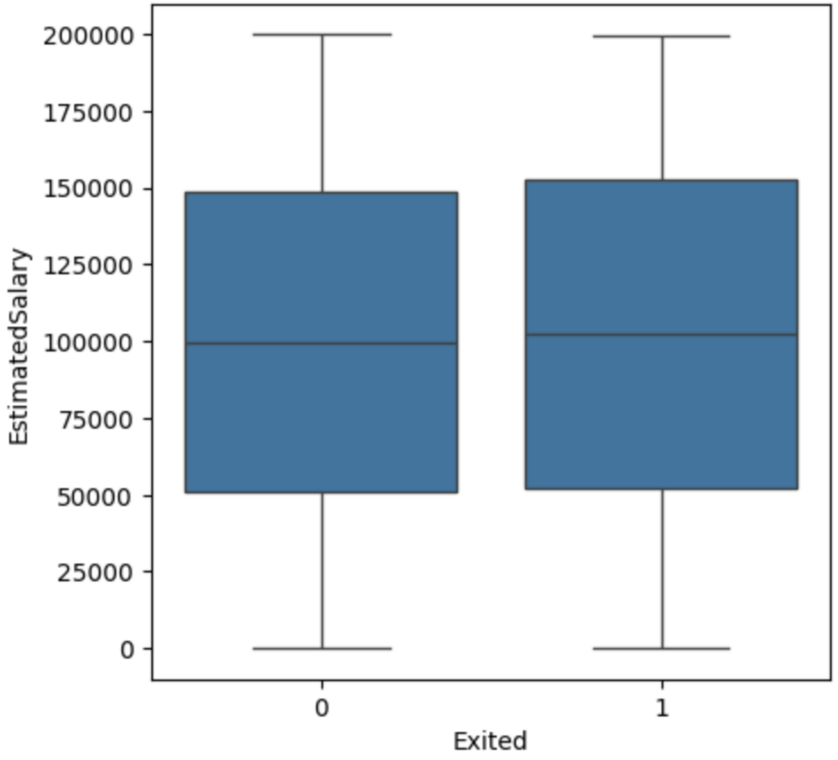


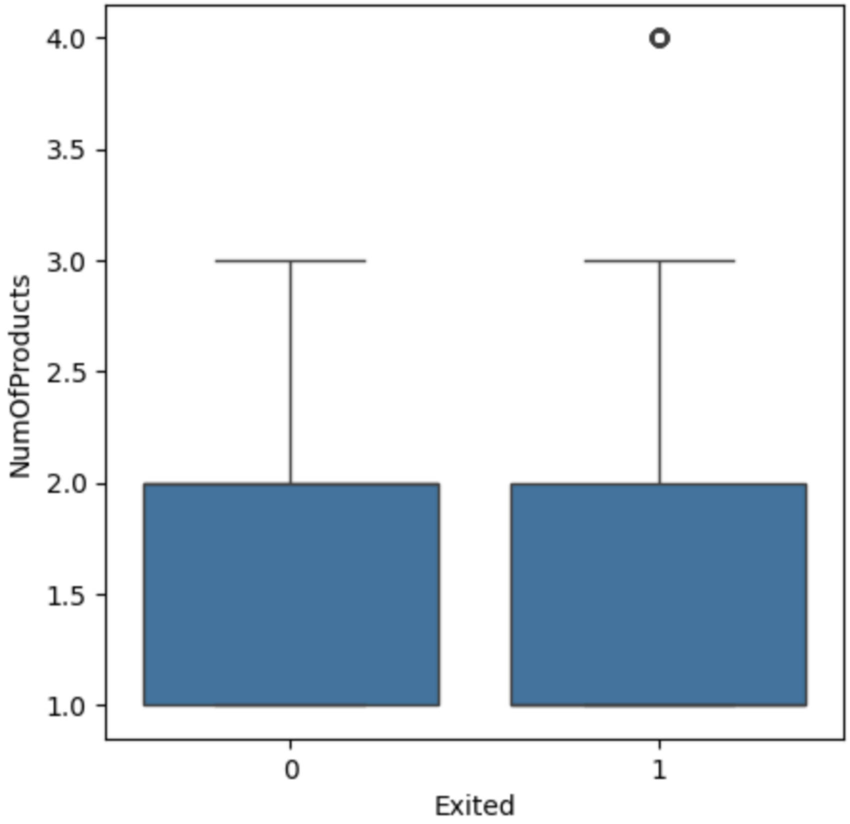
EDA Results – Bivariate Analysis





EDA Results – Bivariate Analysis





Data Preprocessing

IQR analysis for selected features:

No Duplicate Values or Missing Values detected

The outlier analysis based on the IQR reveals the following:

- **CreditScore**: 16 outliers detected.
- **Age**: 411 outliers detected (likely very young or very old customers).
- **Tenure**: No outliers detected.
- **Balance**: No outliers detected.
- **NumOfProducts**: 60 outliers detected.
- **EstimatedSalary**: No outliers detected.

IQR Analysis For Numerical Features

		Q1	Q3	IQR
1	CreditScore	-0.6883585719214899	0.6981093733515285	1.3864679452730182
2	Age	-0.6600184841162697	0.48422460429935044	1.14424308841562
3	Tenure	-0.695981771888179	0.6871298574603482	1.383111629348527
4	Balance	-1.2258476714090278	0.8199204543114416	2.0457681257204694
5	NumOfProducts	-0.9115834940401767	0.8077365626180215	1.7193200566581983
6	EstimatedSalary	-0.8535935281793107	0.8572430923264887	1.7108366205057992

Data Preprocessing

The outlier analysis based on the IQR reveals the following:
CreditScore: 16 outliers detected.
Age: 411 outliers detected (likely very young or very old customers).
Tenure: No outliers detected.
Balance: No outliers detected.
NumOfProducts: 60 outliers detected.
EstimatedSalary: No outliers detected.

We kept all outliers instead of removing them. Keeping the outliers is a valid choice if they are meaningful and not due to data entry errors or anomalies. This ensures that the model learns from the full range of data, including edge cases.

IQR Analysis For Numerical Features

		Q1	Q3	IQR
1	CreditScore	-0.6883585719214899	0.6981093733515285	1.3864679452730182
2	Age	-0.6600184841162697	0.48422460429935044	1.14424308841562
3	Tenure	-0.695981771888179	0.6871298574603482	1.383111629348527
4	Balance	-1.2258476714090278	0.8199204543114416	2.0457681257204694
5	NumOfProducts	-0.9115834940401767	0.8077365626180215	1.7193200566581983
6	EstimatedSalary	-0.8535935281793107	0.8572430923264887	1.7108366205057992

Data Preprocessing (Steps)

1. **Drop Irrelevant Columns**:
    - Remove columns CustomerId, RowNumber and Surname, which don't contribute to the predictive task.
2. **Encode Categorical Variables**:
    - Convert Gender and Geography into numerical representations (e.g., binary or one-hot encoding).
3. **Normalize Numerical Features**:
    - Scale features like CreditScore, Age, Balance, and EstimatedSalary to ensure numerical stability.
4. **Split Dataset**:

- **Training Set**: 8,000 samples with 11 features.
- **Testing Set**: 2,000 samples with 11 features.
- **Handle Class Imbalance** (if necessary):

    1. Analyze the distribution of the target variable (Exited) and apply balancing techniques if required.

Data Preprocessing

1. **Gender**:
    - Since it has only two categories (Male, Female), **binary encoding** was used.
    - This converts Female and Male into 0 and 1, respectively.
2. **Geography**:
    - This column has multiple categories (e.g., France, Spain, Germany), so **one-hot encoding** was used.
    - This creates separate columns like Geography_Germany and Geography_Spain (while dropping one column to avoid redundancy).

Both methods ensure categorical data is transformed into numerical form without introducing unintended biases.

Data Preprocessing (Feature Engineering)

Added Feature Engineering for the existing Bank Churn Dataset

1. **Interaction Features**:
    - Create combinations of features that might have a synergistic effect:
        - Balance-to-Salary Ratio
        - Age-to-Tenure Ratio: (useful if tenure > 0)
2. **Customer Activity**:
    1. Combine IsActiveMember, NumOfProducts, and HasCrCard to create an **Activity Score**:
3. **Age Groups**:
    - Bucket Age into age groups (e.g., 18–25, 26–35, etc.) to capture trends by demographic.
4. **High Credit Risk**:
    - Flag customers with low credit scores (e.g., CreditScore < 500) as a binary feature.

Data Preprocessing (Feature Engineering)

Added Feature Engineering for the existing Bank Churn Dataset

4. Geography Impact:

- Compute average churn rate by geography and add as a feature:
  - For example, if France has a churn rate of 20%, then every French customer gets GeographyChurnRate = 0.20.

6. Product Engagement:

- Flag customers with NumOfProducts > 2 as **Highly Engaged**.

Model Overview Summary

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	188,480
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33

Total params: 190,593 (744.50 KB)
Trainable params: 190,593 (744.50 KB)
Non-trainable params: 0 (0.00 B)

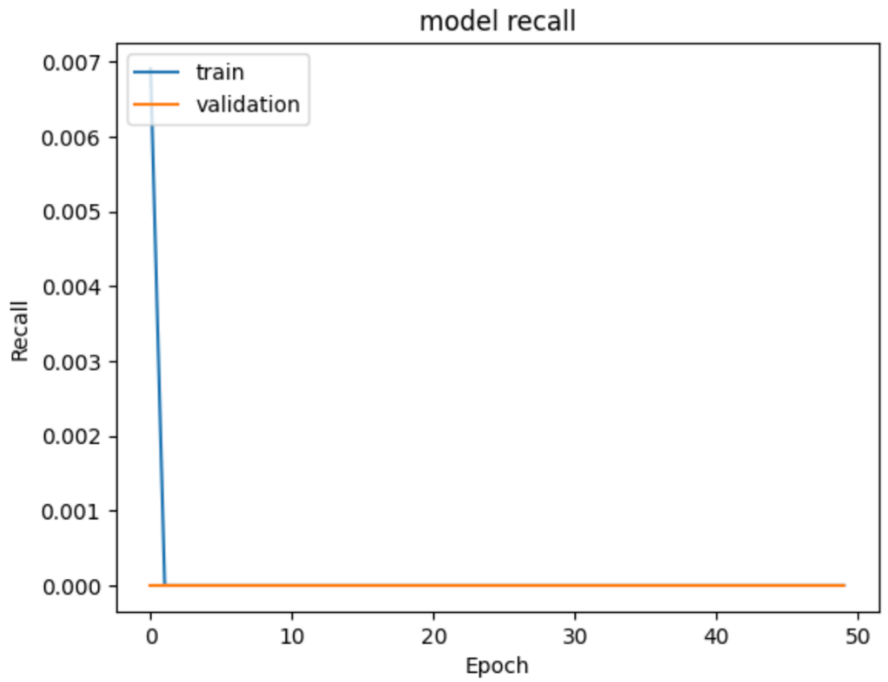
Code Snippet for fitting the Artificial Neural Network:

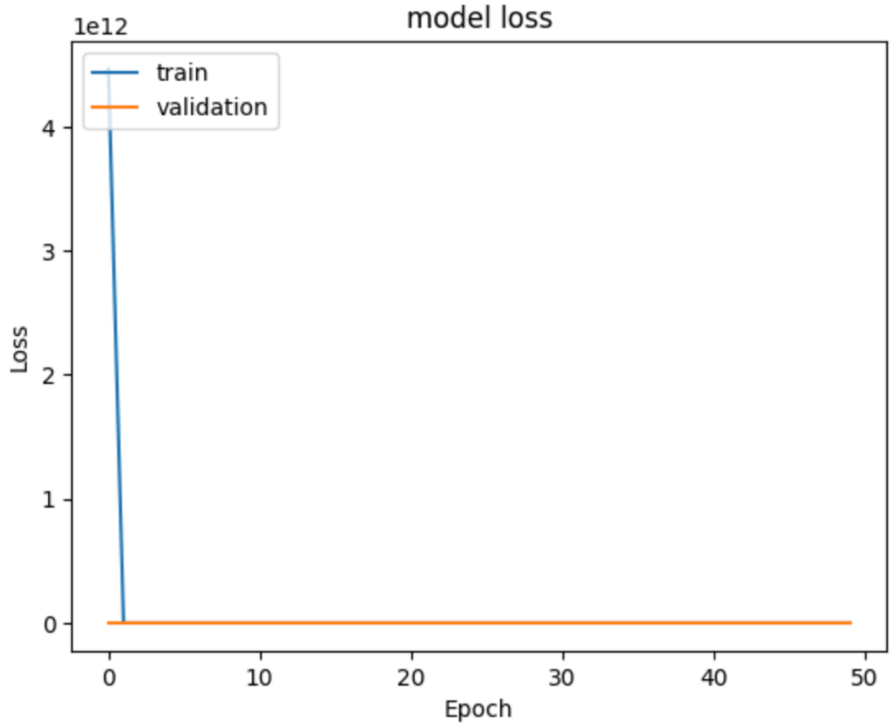
```
# Fitting the ANN

history_0 = model_0.fit(
    X_train, y_train,
    batch_size=32,    ## Complete the code to specify the batch size to use
    validation_data=(X_val,y_val),
    epochs=50,    ## Complete the code to specify the number of epochs
    verbose=1
)
```

```
Epoch 1/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 4s 6ms/step - loss: 21686340550656.0000 - recall: 0.0302 - val_loss: 0.6766 - val_recall: 0.0000e+00
Epoch 2/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 3s 3ms/step - loss: 0.6731 - recall: 0.0000e+00 - val_loss: 0.6614 - val_recall: 0.0000e+00
Epoch 3/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.6586 - recall: 0.0000e+00 - val_loss: 0.6477 - val_recall: 0.0000e+00
Epoch 4/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.6456 - recall: 0.0000e+00 - val_loss: 0.6353 - val_recall: 0.0000e+00
Epoch 5/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.6338 - recall: 0.0000e+00 - val_loss: 0.6241 - val_recall: 0.0000e+00
Epoch 6/50
200/200 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.6231 - recall: 0.0000e+00 - val_loss: 0.6139 - val_recall: 0.0000e+00
```

Model Overview Summary





Model Overview Summary

Model 1: Base Model with Adam Optimizer

•**Objective**: Establish a baseline performance using the Adam optimizer, which is effective at adaptive learning.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Output Layer: 1 neuron, Sigmoid activation (binary classification).

•**Optimizer**: Adam.

•**Loss Function**: Binary Crossentropy.

•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Good initial performance with a focus on faster convergence.

Model	Key Features	Optimizer	Class Balancing	Overfitting Control
Model 1	Base, no balancing or Dropout	Adam	No	No
Model 2	Dropout for overfitting	Adam	No	Yes
Model 3	Balanced with SMOTE	SGD	Yes	No
Model 4	Balanced with SMOTE	Adam	Yes	No
Model 5	Balanced, SMOTE, Dropout	Adam	Yes	Yes

Model Overview Summary

Model 2: Adam Optimizer with Dropout

•**Objective**: Improve the model's generalization ability and reduce overfitting by introducing Dropout layers.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Dropout Layers: Added after each hidden layer (Dropout rate = 0.5).
- Output Layer: 1 neuron, Sigmoid activation.

•**Optimizer**: Adam.

•**Loss Function**: Binary Crossentropy.

•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Improved performance on validation data due to reduced overfitting.

Model	Key Features	Optimizer	Class Balancing	Overfitting Control
Model 1	Base, no balancing or Dropout	Adam	No	No
Model 2	Dropout for overfitting	Adam	No	Yes
Model 3	Balanced with SMOTE	SGD	Yes	No
Model 4	Balanced with SMOTE	Adam	Yes	No
Model 5	Balanced, SMOTE, Dropout	Adam	Yes	Yes

Model Overview Summary

Model 3: Balanced Data with SMOTE + SGD Optimizer

•**Objective**: Address class imbalance using SMOTE to oversample the minority class and train the model with the simpler SGD optimizer.

•**Preprocessing**:

- Apply SMOTE to oversample the minority class.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Output Layer: 1 neuron, Sigmoid activation.

•**Optimizer**: SGD.

•**Loss Function**: Binary Crossentropy.

•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Higher recall due to balanced data but slower convergence compared to Adam.

Model	Key Features	Optimizer	Class Balancing	Overfitting Control
Model 1	Base, no balancing or Dropout	Adam	No	No
Model 2	Dropout for overfitting	Adam	No	Yes
Model 3	Balanced with SMOTE	SGD	Yes	No
Model 4	Balanced with SMOTE	Adam	Yes	No
Model 5	Balanced, SMOTE, Dropout	Adam	Yes	Yes

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	94,240
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 8)	136
dropout_1 (Dropout)	(None, 8)	0
dense_3 (Dense)	(None, 8)	72
dense_4 (Dense)	(None, 1)	9

Total params: 94,985 (371.04 KB)
Trainable params: 94,985 (371.04 KB)
Non-trainable params: 0 (0.00 B)

Model Overview Summary

Model 4: Balanced Data with SMOTE + Adam Optimizer

•**Objective**: Combine the benefits of SMOTE for class balancing with the faster convergence of the Adam optimizer.

•**Preprocessing**:

- Apply SMOTE to oversample the minority class.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Output Layer: 1 neuron, Sigmoid activation.

•**Optimizer**: Adam.

•**Loss Function**: Binary Crossentropy.

•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Better recall and F1-score compared to Model 3, with faster convergence.

Model	Key Features	Optimizer	Class Balancing	Overfitting Control
Model 1	Base, no balancing or Dropout	Adam	No	No
Model 2	Dropout for overfitting	Adam	No	Yes
Model 3	Balanced with SMOTE	SGD	Yes	No
Model 4	Balanced with SMOTE	Adam	Yes	No
Model 5	Balanced, SMOTE, Dropout	Adam	Yes	Yes

Model Overview Summary

Model 5: Balanced Data with SMOTE, Adam Optimizer, and Dropout

•**Objective**: Achieve the best generalization and recall by combining SMOTE, Adam, and Dropout.

•**Preprocessing**:

- Apply SMOTE to oversample the minority class.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Dropout Layers: Added after each hidden layer (Dropout rate = 0.5).
- Output Layer: 1 neuron, Sigmoid activation.

•**Optimizer**: Adam.

•**Loss Function**: Binary Crossentropy.

•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Most balanced model in terms of recall, precision, and generalization.

Model	Key Features	Optimizer	Class Balancing	Overfitting Control
Model 1	Base, no balancing or Dropout	Adam	No	No
Model 2	Dropout for overfitting	Adam	No	Yes
Model 3	Balanced with SMOTE	SGD	Yes	No
Model 4	Balanced with SMOTE	Adam	Yes	No
Model 5	Balanced, SMOTE, Dropout	Adam	Yes	Yes

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	188,480
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 8)	264
dense_3 (Dense)	(None, 1)	9

Total params: 190,833 (745.44 KB)
Trainable params: 190,83
Non-trainable params: 0

```
y_train_pred = model_5.predict(X_train_smote)
#Predicting the results using 0.5 as the threshold
y_train_pred = (y_train_pred > 0.5)
y_train_pred

319/319 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])

[107] y_val_pred = model_5.predict(X_val)
#Predicting the results using 0.5 as the threshold
y_val_pred = (y_val_pred > 0.5)
y_val_pred

50/50 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

Model Overview Summary

Model 5: Balanced Data with SMOTE, Adam Optimizer, and Dropout

- **Objective**: Achieve the best generalization and recall by combining SMOTE, Adam, and Dropout.
- **Preprocessing**:
  - Apply SMOTE to oversample the minority class.
- **Architecture**:
  - Input Layer: Neurons = number of features.
  - Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
  - Dropout Layers: Added after each hidden layer (Dropout rate = 0.5).
  - Output Layer: 1 neuron, Sigmoid activation.
- **Optimizer**: Adam.
- **Loss Function**: Binary Crossentropy.
- **Batch Size**: 32.
- **Epochs**: 50.
- **Outcome**: Most balanced model in terms of recall, precision, and generalization.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	188,480
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 8)	264
dense_3 (Dense)	(None, 1)	9

Total params: 190,833 (745.44 KB)
Trainable params: 190,833 (745.44 KB)
Non-trainable params: 0 (0.00 B)

```
y_train_pred = model_5.predict(X_train_smote)
#Predicting the results using 0.5 as the threshold
y_train_pred = (y_train_pred > 0.5)
y_train_pred
```

```
319/319 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

```
[107] y_val_pred = model_5.predict(X_val)
#Predicting the results using 0.5 as the threshold
y_val_pred = (y_val_pred > 0.5)
y_val_pred
```

```
50/50 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

Model Overview Summary

Model 5: Balanced Data with SMOTE, Adam Optimizer, and Dropout

•**Objective**: Achieve the best generalization and recall by combining SMOTE, Adam, and Dropout.

•**Preprocessing**:

- Apply SMOTE to oversample the minority class.

•**Architecture**:

- Input Layer: Neurons = number of features.
- Hidden Layers: Two layers, 64 and 32 neurons, with ReLU activation.
- Dropout Layers: Added after each hidden layer (Dropout rate = 0.5).
- Output Layer: 1 neuron, Sigmoid activation.

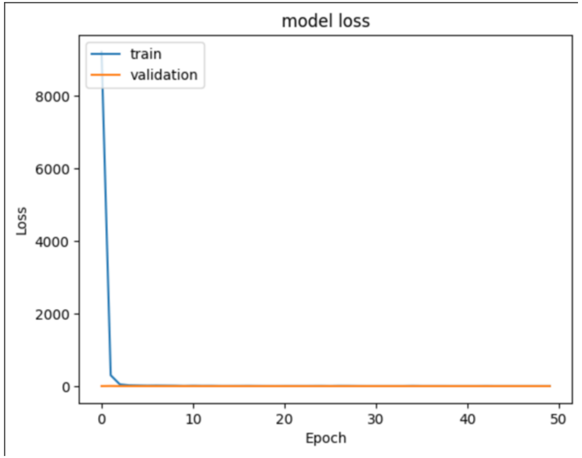
•**Optimizer**: Adam.

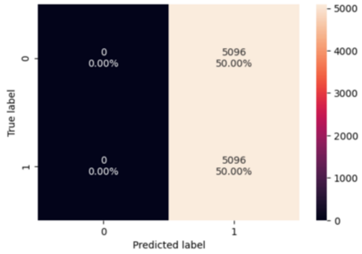
•**Loss Function**: Binary Crossentropy.

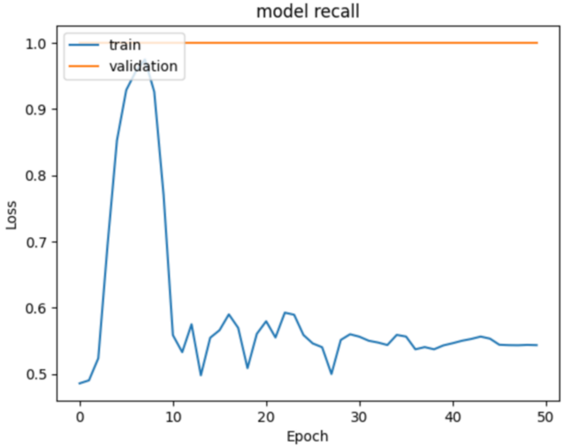
•**Batch Size**: 32.

•**Epochs**: 50.

•**Outcome**: Most balanced model in terms of recall, precision, and generalization.







Model Overview Summary

Model 5: Balanced Data with SMOTE, Adam Optimizer, and Dropout

•**Objective**: Achieve the best generalization and recall by combining SMOTE, Adam, and Dropout.

•Model 5 achieved the best recall and F-1 results (or rather the most robust results) out of all 5 model improvements

•Combining SMOTE and ADAM optimizer helped immensely with class imbalances

```
y_train_pred = model_5.predict(X_train_smote)
#Predicting the results using 0.5 as the threshold
y_train_pred = (y_train_pred > 0.5)
y_train_pred
```

```
319/319 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

```
[107] y_val_pred = model_5.predict(X_val)
      #Predicting the results using 0.5 as the threshold
      y_val_pred = (y_val_pred > 0.5)
      y_val_pred
```

```
50/50 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

Classification report

```
[109] cr=classification_report(y_train_smote,y_train_pred)
      print(cr)
```

```
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00      5096
         1.0       0.50      1.00      0.67      5096

    accuracy                           0.50     10192
   macro avg       0.25      0.50      0.33     10192
weighted avg       0.25      0.50      0.33     10192
```

```
#classification report
cr=classification_report(y_val,y_val_pred)  ## Complete the c
print(cr)
```

```
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00      1274
         1.0       0.20      1.00      0.34       326

    accuracy                           0.20      1600
   macro avg       0.10      0.50      0.17      1600
weighted avg       0.04      0.20      0.07      1600
```

Model Overview Summary (Errors)

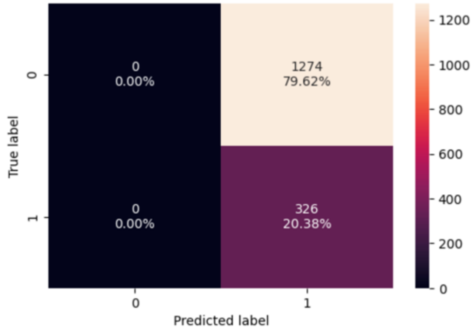
Overfitting to the Training Data

Explanation:

•Overfitting occurs when a model learns the training data too well, including noise and irrelevant patterns, but fails to generalize to unseen data.

•Symptoms:

- Training metrics (e.g., loss, accuracy) are very high, but test metrics remain stagnant.
- Recall, Precision, and other metrics on test data might look identical because the model makes similar predictions regardless of architecture differences.

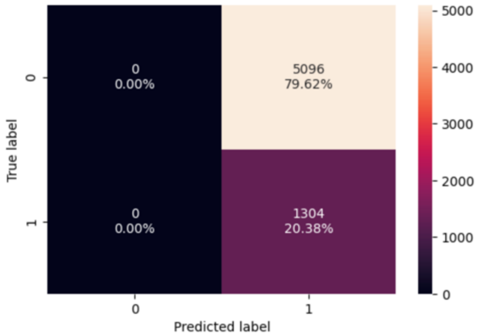


Cause:

•**Insufficient Regularization**: Dropout or L2 regularization may not be applied.

•**Small Dataset**: Models may memorize patterns in the data due to limited training samples.

•**Too Many Parameters**: Complex models (e.g., deep architectures) can overfit if the dataset size is small.



•Result: Confusion Matrices were the same for Models 1, 2 and 4 on both training and test sets

•Loss and Recall reports were not satisfactory to my liking – SMOTE was not enough

Model Overview Summary (Errors)

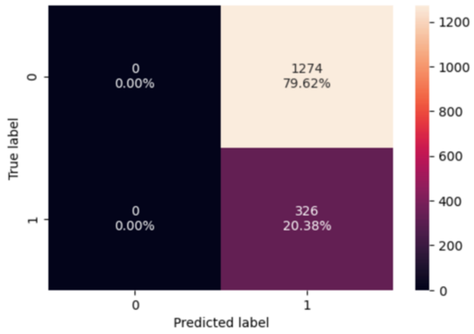
Similar Architectures

Explanation:

•Architectures of models were kept very similar (e.g., same number of layers, neurons, and activation functions), they probably converge to similar decision boundaries.

•Symptoms:

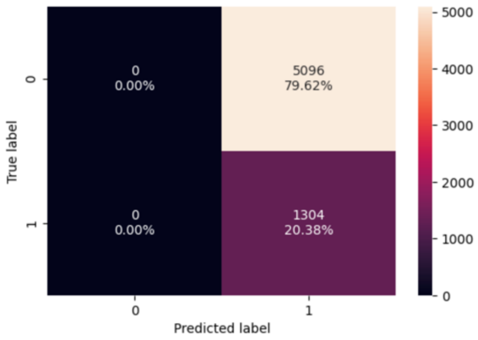
- Nearly identical predictions across models, resulting in similar confusion matrices.



Cause:

•**Minimal Architectural Differences**: Adam vs. SGD optimizers alone may not lead to significant differences if the rest of the architecture is unchanged.

•Result: Confusion Matrices were the same for Models 1, 2 and 4 on both training and test sets



Model Overview Summary (Errors)

How to Address This

For Overfitting

1.Regularization:

1. Add Dropout layers to prevent overfitting.
2. Use L2 regularization on the weights.

2.Increase Data Size:

1. Augment the dataset or gather more samples if possible.

3.Early Stopping:

1. Monitor validation loss and stop training when performance stops improving.

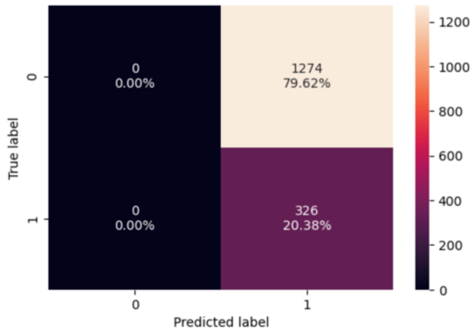
For Validation and Architecture

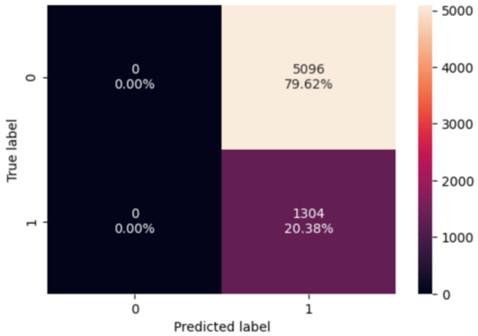
1.Check Data Splits:

1. Ensure stratified splits for balanced class distribution.
2. Avoid data leakage by separating training, validation, and test sets properly.

2.Increase Architectural Variety:

1. Experiment with different numbers of layers, neurons, activation functions, ar optimizers.





APPENDIX

Data Background and Contents

First 5 and last 5 rows of the dataset for reference:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

*Note that RowNumber, CustomerID and Surname were dropped during Data Preprocessing due to being unique but irrelevant values

Statistical Summary and List of Data Types

	count	mean	std	min	25%	50%	75%	max
RowNumber	10000.0	5.000500e+03	2886.895680	1.00	2500.75	5.000500e+03	7.500250e+03	10000.00
CustomerId	10000.0	1.569094e+07	71936.186123	15565701.00	15628528.25	1.569074e+07	1.575323e+07	15815690.00
CreditScore	10000.0	6.505288e+02	96.653299	350.00	584.00	6.520000e+02	7.180000e+02	850.00
Age	10000.0	3.892180e+01	10.487806	18.00	32.00	3.700000e+01	4.400000e+01	92.00
Tenure	10000.0	5.012800e+00	2.892174	0.00	3.00	5.000000e+00	7.000000e+00	10.00
Balance	10000.0	7.648589e+04	62397.405202	0.00	0.00	9.719854e+04	1.276442e+05	250898.09
NumOfProducts	10000.0	1.530200e+00	0.581654	1.00	1.00	1.000000e+00	2.000000e+00	4.00
HasCrCard	10000.0	7.055000e-01	0.455840	0.00	0.00	1.000000e+00	1.000000e+00	1.00
IsActiveMember	10000.0	5.151000e-01	0.499797	0.00	0.00	1.000000e+00	1.000000e+00	1.00
EstimatedSalary	10000.0	1.000902e+05	57510.492818	11.58	51002.11	1.001939e+05	1.493882e+05	199992.48
Exited	10000.0	2.037000e-01	0.402769	0.00	0.00	0.000000e+00	0.000000e+00	1.00

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

Missing Values and Unique Values

Count of Unique Values

	0
RowNumber	10000
CustomerId	10000
Surname	2932
CreditScore	460
Geography	3
Gender	2
Age	70
Tenure	11
Balance	6382
NumOfProducts	4
HasCrCard	2
IsActiveMember	2
EstimatedSalary	9999
Exited	2

dtype: int64

Data does not have missing values before preprocessing
All Data is good before Feature Engineering

Missing Value Treatment

	0
RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	0
IsActiveMember	0
EstimatedSalary	0
Exited	0

dtype: int64

Great Learning

Happy Learning !