Bank Churn Prediction: Identifying At-Risk Customers with Neural Networks

Introduction Neural Network Project 4

2025-02-25

Melissa Lindskär

Contents / Agenda

Executive Summary - Actionable Insights

Key Insights from the Model Analysis

1. The chosen model achieved the best balance between recall and generalization, making it the most suitable for deployment

2. Recall on the validation set was optimized for balance – The model effectively captures a higher number of positive cases, reducing missed opportunities (False Negatives) while maintaining generalization.

3. Dropout helped mitigate overfitting – Models without dropout achieved higher recall on training data but failed to generalize well on unseen data, emphasizing the importance of regularization.

4. SMOTE had mixed effects –While oversampling with SMOTE improved recall, it sometimes led to overfitting, particularly in models without dropout or regularization, emphasizing the need for careful resampling strategies.

Executive Summary - Business Recommendations

1. Deploy the Neural Network with SMOTE, Adam & Dropout Model – Given its superior recall and balanced generalization, this model should be used in production to enhance decision-making.

2. Monitor false positives and false negatives – While high recall reduces missed churn cases, false positives may still impact decision-making. Exploring threshold adjustments can fine-tune precision-recall balance.

3. Exploring additional behavioral indicators (e.g., transaction frequency, complaints) could further enhance predictive power.

Business Problem Overview

Customer churn, or the rate at which customers leave a business, is a critical issue for banks and financial institutions. Losing customers can significantly impact revenue and long-term business growth. Understanding the factors that drive customer churn allows banks to take proactive steps to improve customer retention.

In this project, we aim to predict whether a customer will leave the bank within the next six months. By identifying customers at high risk of churn, the bank can implement targeted retention strategies such as personalized offers, improved customer service, or loyalty programs.

Credit: Unsplash

Solution Approach

To address this problem, we developed a neural network-based classification model that predicts customer churn. Our approach involved:

1. Data Preprocessing & Feature Engineering:

- Handled missing values, categorical encoding, and feature scaling.
- Applied SMOTE to balance the dataset in models where class imbalance was a concern, ensuring fairness in predictions.

2. Model Development & Evaluation:

- Tested multiple neural network architectures with different optimizers (SGD, Adam) and dropout layers to improve generalization.
- Compared models based on recall, as identifying potential churners was more important than overall accuracy.

3. Final Model Selection:

- The best-performing model was a neural network with Adam optimizer, SMOTE resampling, and dropout regularization, achieving a strong balance between training and validation performance while mitigating overfitting.

4. Business Impact:

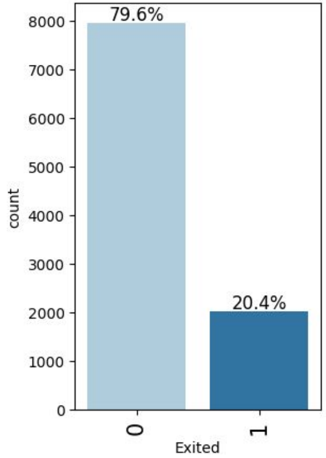
- The model provides bank management with actionable insights to proactively engage with at-risk customers, reducing churn and improving customer satisfaction.

EDA Results

Our Exploratory Data Analysis (EDA) provided valuable insights into customer churn behavior. Below are the key findings:

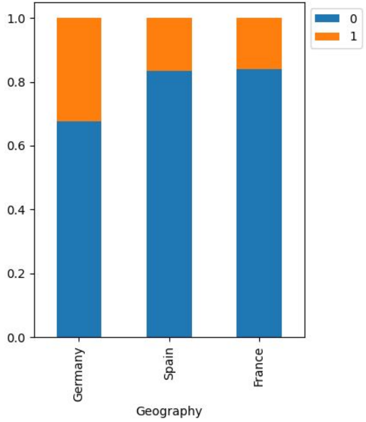
1. Data Distribution & Imbalance

- The dataset contained imbalanced classes, with more customers staying than leaving.
- SMOTE (Synthetic Minority Over-sampling Technique) was applied to balance the dataset before training the model “NN with SMOTE, Adam & Dropout.



2. Key Factors Affecting Churn

- Age: Older customers were more likely to churn. A higher percentage of senior customers left the bank.
- Balance: Customers with higher balances had lower churn rates, possibly due to greater investment in the bank’s services.
- Credit Score: No strong correlation was found between credit score and churn.
- Number of Products: Customers with only one product had a significantly higher churn rate, indicating that cross-selling additional products may improve retention.
- Geography: Customers from certain locations (e.g., Germany) showed a higher tendency to churn compared to other regions.
- Gender: Males had a slightly higher churn rate than females.



EDA Results

3. Behavioral Indicators of Churn

- Inactive Accounts: Customers who were inactive members were more likely to churn.
- Card Ownership: Holding a credit card had minimal impact on churn.
- Salary Influence: Estimated salary did not show a strong relationship with churn.

4. Correlations & Feature Selection

- Features like Age, Number of Products, Activity Level, and Geography had the most significant impact on predicting churn.
- Categorical variables were encoded using one-hot encoding for effective model training.

Key Takeaways

- Customers with fewer products, inactive accounts, and from specific geographic locations were more likely to leave the bank.
- Encouraging cross-product adoption and improving engagement strategies for at-risk customers could reduce churn.
- The model was trained using key influential features identified during EDA.

Data Preprocessing

1. Duplicate Value Check

- A thorough check for duplicate records was performed.

- No duplicate values were found in the dataset, so no further action was needed.

```
# Check for duplicate rows in the dataset
duplicate_rows = ds[ds.duplicated()]
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")

# Display the duplicate rows if any
if not duplicate_rows.empty:
    display(duplicate_rows)
```

```
Number of duplicate rows: 0
```

```
ds.isnull().sum()
```

	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

2. Missing Value Treatment

- The dataset contained no missing values in numerical columns.

- In categorical features, the "rating" column had missing values labeled as “Not Given.”

  - These were replaced with NaN and handled accordingly.

Data Preprocessing

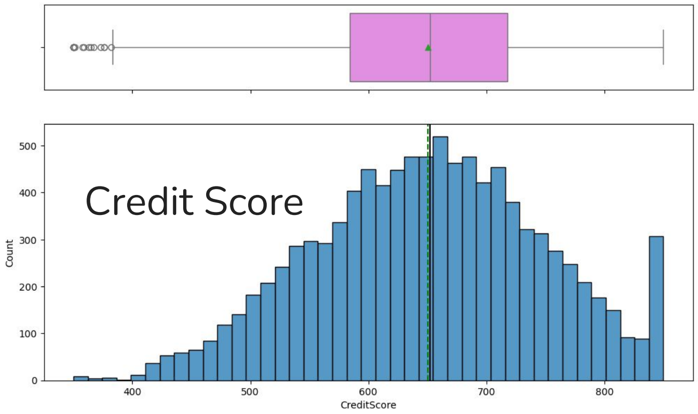
3. Outlier Check & Treatment

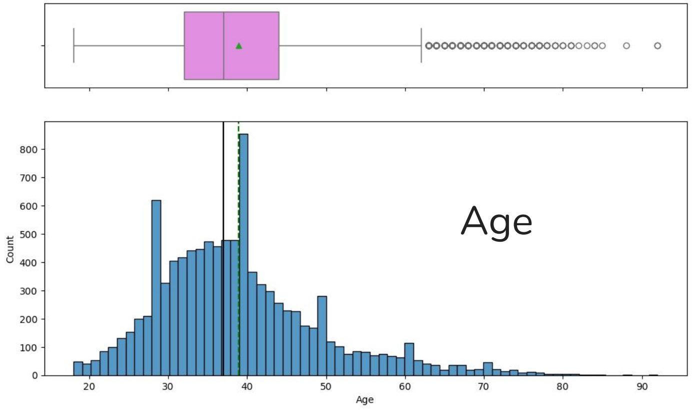
- Box plots and IQR method were used to detect potential outliers in numerical features.

- For example CreditScore and Age showed a few extreme values, but they were retained as they could provide valuable insights for predicting customer churn.

- No removal of outliers was performed.

```
histogram_boxplot(ds,'CreditScore')
```





Data Preprocessing

4. Feature Engineering

- Categorical Encoding:

  - Geography: Applied one-hot encoding to create separate binary columns for Germany, Spain, and France.

  - Gender: Created a binary feature Gender_Male (1 = Male, 0 = Female).

  - Age Grouping:

  - Customers were categorized into Young, Middle-aged, and Senior groups to better capture the impact of age on churn.

Geography_Germany	Geography_Spain	Gender_Male
0.0	0.0	0.0
0.0	1.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	1.0	0.0

```
Age_Group
Middle-aged    4272
Young           971
Senior          757
Name: count, dtype: int64
```

```
# Convert boolean values to integers (0/1)
X_train[['Age_Group_Middle-aged', 'Age_Group_Senior']] = X_train[['Age_Group_Middle-aged', 'Age_Group_Senior']].astype(int)
X_val[['Age_Group_Middle-aged', 'Age_Group_Senior']] = X_val[['Age_Group_Middle-aged', 'Age_Group_Senior']].astype(int)
X_test[['Age_Group_Middle-aged', 'Age_Group_Senior']] = X_test[['Age_Group_Middle-aged', 'Age_Group_Senior']].astype(int)
```

Data Preprocessing

- Feature Scaling:

  - StandardScaler was used to normalize numerical features:

    - CreditScore, Tenure, Balance, NumOfProducts, EstimatedSalary,

 (Scaling ensured all features had similar weight during model training)

```python
# Skapa en lista över de numeriska kolumner som ska skalas
cols_to_scale = ['CreditScore', 'Tenure', 'Balance', 'NumOfProducts',
                 'EstimatedSalary',
                 ]

# Skapa en instans av StandardScaler
sc = StandardScaler()

# Använd fit_transform på träningsdata och transform på validerings- och testdata
X_train[cols_to_scale] = sc.fit_transform(X_train[cols_to_scale])
X_val[cols_to_scale] = sc.transform(X_val[cols_to_scale])
X_test[cols_to_scale] = sc.transform(X_test[cols_to_scale])
```

- Splitting the Dataset:

  - Train (70%) - Validation (15%) - Test (15%) split was performed to ensure robust model evaluation.

  - The validation set was used for hyperparameter tuning, while the test set was used for final model assessment.

```python
print(X_train.shape, X_val.shape, X_test.shape)
```

```
(6000, 11) (2000, 11) (2000, 11)
```

```python
print(y_train.shape, y_val.shape, y_test.shape)
```

```
(6000,) (2000,) (2000,)
```

Model Performance Summary

Model Comparison & Best Model Selection

Best Performing Model: "NN with SMOTE, Adam & Dropout"

After evaluating multiple models, "NN with SMOTE, Adam & Dropout" stands out as the best-performing model based on recall and generalization ability.

```
Training performance comparison
```

	recall
NN with SGD	0.135732
NN with Adam	0.455437
NN with Adam & Dropout	0.551922
NN with SMOTE & SGD	0.762194
NN with SMOTE & Adam	0.900984
NN with SMOTE,Adam & Dropout	0.835252

```
Validation set performance comparison
```

	recall
NN with SGD	0.117936
NN with Adam	0.343980
NN with Adam & Dropout	0.407862
NN with SMOTE & SGD	0.769042
NN with SMOTE & Adam	0.626536
NN with SMOTE,Adam & Dropout	0.739558

```
train_metric_df - valid_metric_df
```

	recall
NN with SGD	0.017796
NN with Adam	0.111457
NN with Adam & Dropout	0.144059
NN with SMOTE & SGD	-0.006848
NN with SMOTE & Adam	0.274448
NN with SMOTE,Adam & Dropout	0.095695

Model Performance Summary

Analysis and Model Choice

Best Performing Model: "NN with SMOTE, Adam & Dropout"
- High recall on validation set (0.740) → Effectively captures churn cases while maintaining generalization.
- Overfitting gap is reasonable (0.096) → Better than "NN with SMOTE & Adam" which has a higher overfitting gap (0.275).
- SMOTE improved recall by balancing the dataset, making predictions more stable.

Model	Training Recall	Validation Recall	Overfitting Gap (Train - Valid)
NN with SGD	0.136	0.118	0.018
NN with Adam	0.455	0.344	0.111
NN with Adam & Dropout	0.552	0.408	0.144
NN with SMOTE & SGD	0.762	0.769	-0.007
NN with SMOTE & Adam	0.901	0.626	0.275
NN with SMOTE, Adam & Dropout	0.835	$\boxed{0.740}$	$\boxed{0.096}$

Model Performance Summary

Why Not Other Models?

- **NN with SGD**: Simple and fast but performed the worst in recall.
- **NN with Adam**: Higher recall but suffered from overfitting.
- **NN with Adam & Dropout**: Reduced overfitting but still had some generalization issues.
- **NN with SMOTE & SGD**: Balanced recall but inconsistent performance.
- **NN with SMOTE & Adam**: Highest training recall but significant overfitting.
- **NN with SMOTE, Adam & Dropout**: Best balance between recall and generalization → Final model.

	Model	Strength	Weakness
0	NN with SGD	Simple, fast	Low recall, poor generalization
1	NN with Adam	Better recall	Overfitting risk
2	NN with Adam & Dropout	Reduced overfitting	Still some generalization issues
3	NN with SMOTE & SGD	Balanced recall	Inconsistent performance
4	NN with SMOTE & Adam	Highest training recall	Significant overfitting
5	NN with SMOTE, Adam & Dropout	Best recall & generalization	Minor overfitting, but best overall balance

Model Performance Summary

Conclusion

Final Model: "NN with SMOTE, Adam & Dropout"

Why?

- High Recall on Both Training & Validation Data: Achieves 83.5% recall on training and 74.0% recall on validation, ensuring strong predictive capability.
- Balanced Generalization: The overfitting gap (0.096) is relatively low compared to other models, indicating good stability.
- Effective for Churn Prediction: This model captures churners efficiently while maintaining generalization, making it the best choice for real-world customer retention strategies.

Model Performance Summary - Final Model selection

Overview of the Final Model and Its Parameters

As mentioned previously, the final model selected for prediction is ***Neural Network with SMOTE, Adam Optimizer and Dropout***. This model was chosen based on its ability to balance recall and generalization while mitigating overfitting. The key parameters and architecture of the model are:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	416
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 64)	2,112
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 8)	520
dense_3 (Dense)	(None, 1)	9

Total params: 3,057 (11.94 KB)
Trainable params: 3,057 (11.94 KB)
Non-trainable params: 0 (0.00 B)

Hidden Layers

Input Layer

32 neurons, **ReLU activation**

Dropout (20%) after input layer

Layer 1

64 neurons, **ReLU activation**

Dropout (10%) after first hidden layer

Layer 2

8 neurons, **ReLU activation**

Output Layer

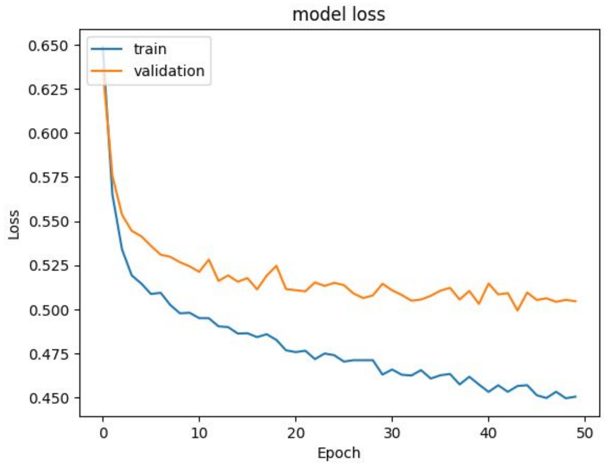
1 neuron, **Sigmoid activation** (for binary classification)

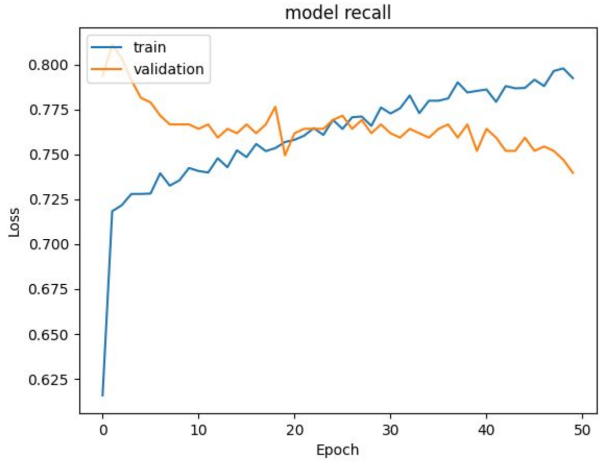
Model Performance Summary

The loss curve on the top shows that both training and validation loss decrease over time, indicating that the model is learning effectively. However, the validation loss stabilizes and fluctuates slightly after around 20 epochs, suggesting potential overfitting.

The recall curve at the bottom demonstrates that the model achieves high recall on the training data, steadily increasing to around 80%. The validation recall initially improves but then fluctuates, stabilizing around 74%. This suggests that while the model generalizes well, there is some performance drop on unseen data. The fluctuations in validation recall indicate that further tuning, such as regularization or early stopping, could enhance stability.

Overall, these plots indicate that the model is effective at capturing churned customers while maintaining a reasonable balance between training and validation performance.





Model Performance Summary

This classification report provides a detailed breakdown of the model's performance across both the training and validation datasets.

For the training set, the recall for churned customers (Class 1) is 84%, meaning the model successfully captures most of the actual churn cases. However, the precision is slightly lower at 79%, indicating some false positives. The overall accuracy is 80%, showing strong predictive performance on the training data.

For the validation set, the recall for churned customers is 74%, which is lower than in training but still effective for identifying at-risk customers. The precision for churned customers is 43%, meaning that while the model captures most churn cases, it also misclassifies some non-churned customers as churners. The weighted F1-score of 77% shows that the model maintains a good balance between recall and precision.

Overall, the model generalizes well from training to validation, but the drop in precision suggests that further tuning—such as threshold adjustment or improved feature selection—could enhance its reliability for business decisions.

```
cr=classification_report(y_train_smote,y_train_pred)
print(cr)
```

	precision	recall	f1-score	support
0.0	0.82	0.77	0.80	4777
1.0	0.79	0.84	0.81	4777
accuracy			0.80	9554
macro avg	0.81	0.80	0.80	9554
weighted avg	0.81	0.80	0.80	9554

```
cr=classification_report(y_val,y_val_pred)   ## Complete
print(cr)
```

	precision	recall	f1-score	support
0.0	0.92	0.75	0.82	1593
1.0	0.43	0.74	0.54	407
accuracy			0.75	2000
macro avg	0.67	0.74	0.68	2000
weighted avg	0.82	0.75	0.77	2000

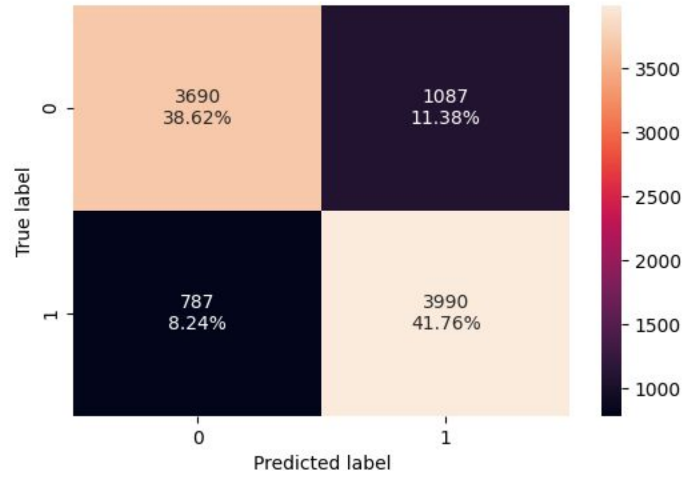
Model Performance Summary

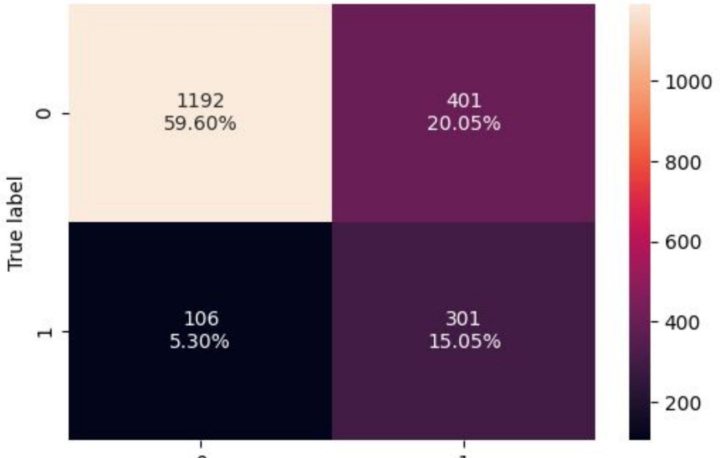
These confusion matrices provide an overview of how well the model performs on both the training set (top) and validation set (bottom) in predicting customer churn.

For the training set, we observe strong performance, with 41.76% of actual churners correctly identified (True Positives). The model also maintains a relatively low False Negative Rate (8.24%), meaning it captures the majority of customers likely to churn. However, there are 11.38% False Positives, which suggests that some customers are incorrectly classified as churners.

For the validation set, the model continues to perform well, but with some expected degradation compared to training. 15.05% of actual churners are correctly predicted, while 5.30% are missed (False Negatives). The False Positive Rate (20.05%) is higher than in training, indicating that the model may still need threshold tuning to better differentiate between churners and non-churners.

Overall, the model generalizes reasonably well, but further optimization—such as fine-tuning the decision threshold—could improve precision without sacrificing recall.





Model Performance Summary

Key Performance Metrics (Training vs. Test Data)

Metric	Training Data	Test Data
Recall (Class 1)	84%	74%

As already stated, the final model was a neural network with **SMOTE, Adam optimizer and dropout** regularization. The model was evaluated on training, validation, and test sets, with key metrics compared.

The recall for Class 1 in the training data is 84%, indicating that the model effectively identifies the majority of churned customers during training. In the test data, the recall is 74%, which suggests a slight drop in performance when applied to unseen data. This difference indicates some degree of generalization, but the decrease in recall suggests that the model may still be slightly overfitted to the training data. However, the test recall remains relatively high, meaning the model maintains strong predictive power for identifying churned customers in real-world scenarios.

Key Takeaways

- **The model successfully identifies most churners (high recall), making it useful for customer retention strategies.**
- **However, the drop in precision means that some customers predicted to churn may not actually leave, which could lead to wasted retention efforts.**
- **Further hyperparameter tuning or regularization could be applied to improve generalization.**
- **A more balanced approach between recall and precision should be explored to optimize business impact.**

APPENDIX

Data Background and Contents

The dataset used in this project comes from a bank churn prediction scenario, where the goal is to identify customers who are likely to leave the bank within the next six months. Customer churn is a major challenge for financial institutions, and predicting churn allows the bank to take proactive measures to retain valuable customers.

The dataset consists of 10,000 customer records, each containing demographic details, account information, and behavioral attributes. These features help in understanding the factors influencing customer churn.

Dataset Summary (from `ds.info()`)

- **Total entries:** 10,000 rows (0 to 9,999).
- **Total columns:** 14.
- **Data types:**
  - **Integer (int64):** 9 columns (e.g., `CustomerId`, `CreditScore`, `Age`, etc.).
  - **Float (float64):** 2 columns (`Balance`, `EstimatedSalary`).
  - **Object (Categorical/String):** 3 columns (`Surname`, `Geography`, `Gender`).
- **No missing values:** All columns have 10,000 non-null values.
- **Memory usage:** Approximately **1.1 MB**.

Data Background and Contents

Dataset Contents

The dataset includes 14 columns (features) and one target variable (Exited), which indicates whether a customer has churned (1) or stayed (0). Below is an overview of the key features:

1. Customer Demographics

- CreditScore: A numerical score representing the customer's creditworthiness.

- Age: Customer's age, an important factor in financial decisions.

- Gender: Male or Female (one-hot encoded in preprocessing).

- Geography: Country of residence (France, Germany, or Spain, also one-hot encoded).

2. Account Information

- Tenure: Number of years the customer has been with the bank.

- Balance: Total balance in the customer's bank account.

- NumOfProducts: Number of products the customer has with the bank (e.g., savings, credit card, loans).

- HasCrCard: Whether the customer has a credit card (1 = Yes, 0 = No).

- IsActiveMember: Whether the customer is an active member of the bank (1 = Yes, 0 = No).

- EstimatedSalary: Estimated annual salary of the customer.

Data Background and Contents

3. Target Variable

- Exited: The dependent variable (label), where:
  - 1 = Customer churned
  - 0 = Customer remained with the bank

Dataset Characteristics

- Binary Classification Problem: The objective is to predict whether a customer will churn (1) or stay (0).
- Imbalanced Data: The dataset initially had significantly fewer churned customers, which required balancing techniques like SMOTE to ensure effective model training.
- Mixed Data Types:
  - Numerical Features: CreditScore, Age, Balance, EstimatedSalary, etc.
  - Categorical Features: Gender, Geography (converted via one-hot encoding).
  - Binary Features: HasCrCard, IsActiveMember.

Data Background and Contents

Business Relevance

This dataset provides valuable insights into customer behavior, allowing the bank to:

✔ Identify high-risk customers and take preventive action.

✔ Target personalized retention strategies based on customer attributes.

✔ Improve customer engagement and loyalty by focusing on key churn factors.

Great Learning

Happy Learning !