Neural Networks Project

Bank Churn Prediction

Isaac Gross - May 16, 2025

Contents / Agenda

- Executive Summary
- Business Problem
- Solution Approach
- Exploratory Data Analysis
- Data Preparation
- Model Building & Improvement
- Model Comparison
- Final Selection
- Business Recommendations & Takeaways

Executive Summary

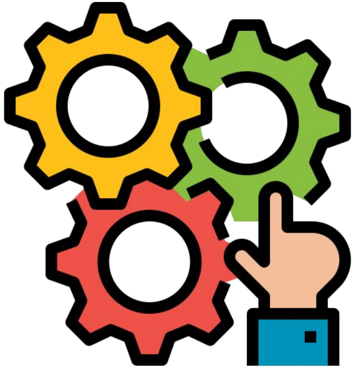
Our bank is losing customers each month, and manual retention strategies aren't scalable.

We built a deep learning model to help flag customers most likely to churn in the next 6 months, so retention teams can focus their efforts where it matters most.

After evaluating multiple neural network architectures, the best performer was a model trained on SMOTE-balanced data, optimized using Adam, and regularized with Dropout.

It delivered a strong recall on churn cases while maintaining precision:

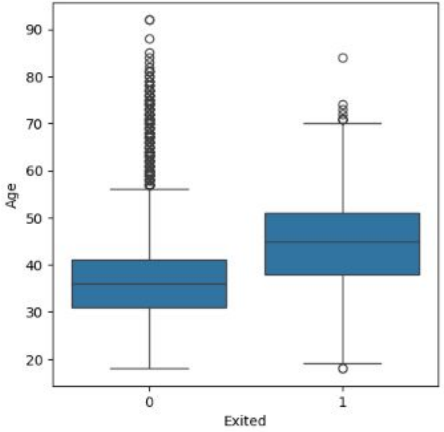
- **F1 Score: 58%**
- **Recall: 77%**
- **Precision: 47%**

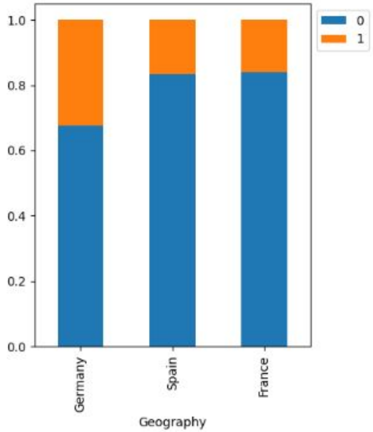


Executive Summary

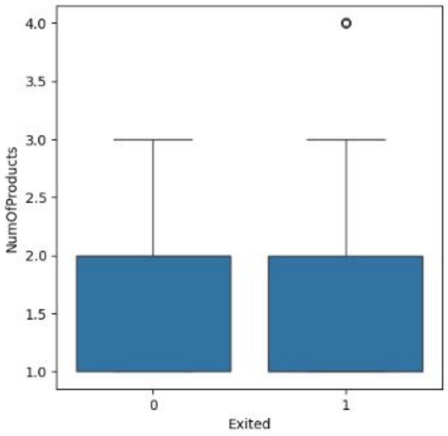
Churners are more likely to be:

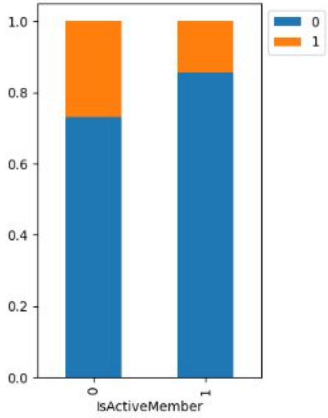
- From Germany
- Inactive members
- Aged 45+
- Holding high balances but using fewer products





Customers with 2+ products and regular activity are most likely to stay

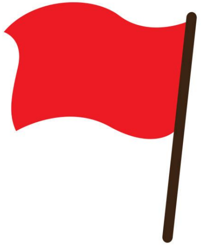




Executive Summary

Recommendations:

- Use this model to flag high churn risk accounts for early intervention
- Integrate predictions into your CRM or retention pipeline to trigger offers, outreach, or reviews
- Retrain the model quarterly as new churn data is collected
- Explore explainability to understand top drivers in real time



By prioritizing recall, we ensure fewer high-risk customers are missed — enabling faster, targeted retention strategies and minimizing revenue loss from preventable churn.

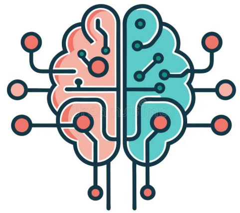
Business Problem

Customer churn is a recurring challenge in the banking sector. Retaining existing customers is far more cost-effective than acquiring new ones, but identifying which customers are at risk of leaving is difficult to scale manually.

The bank needs a way to proactively identify customers who are likely to exit in the next 6 months so retention efforts can be targeted more effectively.

Challenges:

- The data is imbalanced: churned customers account for just ~20% of the base.
- Traditional methods miss subtle behavioral patterns across geography, product usage, and account activity.
- Manual rules and heuristics are inconsistent and reactive.



Solution Approach

We developed a classification model using a neural network trained on customer behavior, demographic, and engagement data.

To handle imbalance and improve generalization, we:

- Applied SMOTE to synthetically balance churned customers in training
- Tuned multiple architectures with SGD and Adam optimizers
- Added Dropout layers to reduce overfitting

The final model helps flag churn risk early, allowing the business to retain more customers with less effort.

EDA Results: Key Findings

- **Churned customers** made up ~20% of the dataset
- **Germany** had the highest churn rate (~32%), while **France** had the lowest (~16%)
- **Inactive members** were significantly more likely to churn
- Customers with **only one product** had a noticeably higher churn rate than those with two
- **Older customers (45+)** churned at higher rates than younger segments
- **High account balance** was not protective — even customers with large balances were leaving
- Gender had a minor effect: **females churned slightly more often**
- No strong correlations between features, suggesting nonlinear interactions best captured by neural networks

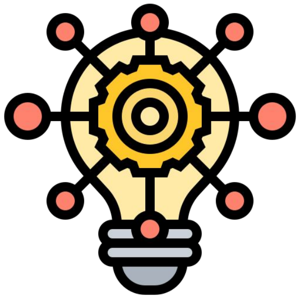
EDA Results: Insights

✅ **Does geography affect churn?**

Yes — customers from Germany had a significantly higher likelihood of churning than other regions.

✅ **Does activity level matter?**

Yes — inactive members churned at a much higher rate than active members.



✅ **Are older customers more likely to churn?**

Yes — churn rate increased steadily with age, especially past 45.

✅ **Does product usage affect churn?**

Yes — customers with only one product were far more likely to churn than those with two.

✅ **Does account balance influence churn?**

Surprisingly, no — high balances did not reduce churn risk.

Data Preprocessing

We applied a series of transformations to prepare the data for neural network modeling:

✔️ **Feature Selection & Cleaning** - No duplicate records found & Removed irrelevant columns

✔️ **Categorical Encoding** - One-hot encoded Geography & binary encoded Gender; dropped rows with missing values

✔️ **Scaling** - Standardized numerical features & fixed negative values

✔️ **Data Splitting** - Stratified split into 63% Train, 7% Validation, 30% Test; applied all transformations consistently

✔️ **Imbalance Handling** - Used SMOTE to oversample churn cases (Exited = 1) in training; also tested undersampling

Model Building - Baseline (SGD Optimizer)

We started with a basic neural network trained using the **SGD (Stochastic Gradient Descent)** optimizer to establish a performance baseline.

✔️ **Architecture** – 2 hidden layers

- → 64 neurons → 32 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – SGD with learning rate 0.001

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall, to prioritize catching churners

Performance (Validation Set):

- **Recall:** 1.4%
- **Precision:** 60.0%
- **F1 Score:** 2.7%
- **Accuracy:** ~80% (but misleading due to class imbalance)

The model failed to learn patterns from minority class (Exited = 1).
It defaulted to predicting the majority class, resulting in extremely low recall.

Model Improvement – Adam Optimizer

We replaced SGD with the **Adam optimizer**, which uses adaptive learning rates for faster and more stable convergence.

✔️ **Architecture** – 2 hidden layers

- → 64 neurons → 32 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – Adam

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall

Performance (Validation Set):

- **Recall:** 46.8%
- **Precision:** 66.0%
- **F1 Score:** 55.0%
- **Accuracy:** 84%

Switching to Adam significantly improved recall from 1.4% to 46.8%.
The model was now able to identify nearly half of the churned customers.

Model Improvement – Adam + Dropout

We introduced **Dropout layers** to reduce overfitting and improve generalization.

✔️ **Architecture** – 3 hidden layers

- → 32 neurons → Dropout(0.2) → 16 neurons → Dropout(0.1) → 8 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – Adam

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall

Performance (Validation Set):

- **Recall:** 49.6%
- **Precision:** 76.2%
- **F1 Score:** 60.4%
- **Accuracy:** 86%

Dropout helped reduce overfitting and slightly improved recall and F1 score.

Model Improvement – SMOTE + SGD

We applied **SMOTE** to balance the churn class in the training data and retrained using **SGD**.

✔️ **Architecture** – 2 hidden layers

- → 64 neurons → 16 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – SGD

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall

Performance (Validation Set):

- **Recall:** 76.2%
- **Precision:** 39.0%
- **F1 Score:** 51.6%
- **Accuracy:** 70.6%

Balancing the data had a major impact on recall, but precision dropped.

Model Improvement – SMOTE + Adam

Next, we used **SMOTE** with the more stable **Adam optimizer**.

✔️ **Architecture** – 2 hidden layers

- → 64 neurons → 32 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – Adam

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall

Performance (Validation Set):

- **Recall:** 68.5%
- **Precision:** 45.3%
- **F1 Score:** 54.5%
- **Accuracy:** 78%

Good generalization and better stability compared to SMOTE + SGD.

Model Improvement – SMOTE + Adam + Dropout

Our final model combined **SMOTE**, **Adam optimizer**, and **Dropout** for regularization.

✔️ **Architecture** – 2 hidden layers

- → 32 neurons → Dropout(0.2) → 16 neurons → Dropout(0.1) → 8 neurons → Output
- Activation: ReLU (hidden), Sigmoid (output)

✔️ **Optimizer** – Adam

✔️ **Loss Function** – Binary Crossentropy

✔️ **Evaluation Metric** – Recall

Performance (Validation Set):

- **Recall:** 78.3% ✅
- **Precision:** 46.0%
- **F1 Score:** 58.0%
- **Accuracy:** 77%

*This model achieved the **best recall** with **stable validation** performance.*

Model Comparison

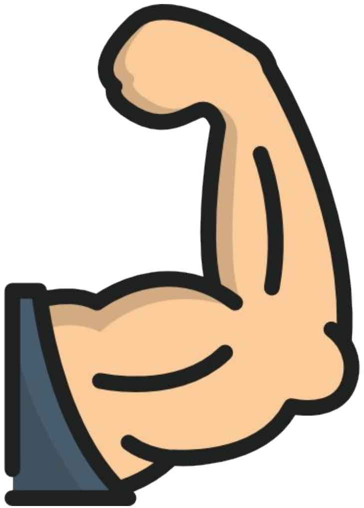
We compared six neural network models using **Recall** as the primary metric, due to its importance in identifying churned customers. Below is a summary of each model's validation performance:

Model	Optimizer	SMOTE	Dropout	Recall	Precision	F1 Score	Accuracy
Baseline (SGD)	SGD	❌	❌	**1.4%**	60.0%	2.7%	80%
Adam	Adam	❌	❌	**46.8%**	66.0%	55.0%	84%
Adam + Dropout	Adam	❌	✅	**49.6%**	76.2%	60.4%	86%
SMOTE + SGD	SGD	✅	❌	**76.2%**	39.0%	51.6%	71%
SMOTE + Adam	Adam	✅	❌	**68.5%**	45.3%	54.5%	78%
✅ Final Model: SMOTE + Adam + Dropout	Adam	✅	✅	**78.3%**	46.0%	58.0%	77%

Final Selection

📌 **Neural Network with SMOTE + Adam + Dropout**

- Best recall (78.3%)
- Strong generalization
- Balanced performance across key metrics



Prioritized recall to minimize missed churners and support early intervention strategies.

Business Recommendations & Takeaways

✅ **Use the model to flag at-risk customers**

- Deploy the final neural network model in the CRM or retention workflow
- Prioritize customers predicted to churn for proactive outreach (offers, support, incentives)

✅ **Act early on churn signals**

- The model successfully identifies ~77% of churners in advance
- Enables retention teams to engage customers before they exit

Business Recommendations & Takeaways

✅ **Monitor and retrain quarterly**

- Retrain the model with updated data every 3–6 months
- Ensure performance stays consistent as customer behavior evolves

✅ **Enable explainability**

- Add tools to understand top churn drivers for individual predictions
- Equip business teams with clear insights to guide targeted actions

By catching churn early, we can reduce customer loss, improve retention strategy efficiency, and drive long-term revenue growth.

Business Recommendations & Takeaways

✅ **Monitor and retrain quarterly**

- Retrain the model with updated data every 3–6 months
- Ensure performance stays consistent as customer behavior evolves

✅ **Enable explainability**

- Add tools to understand top churn drivers for individual predictions
- Equip business teams with clear insights to guide targeted actions

Prioritizing recall allows us to catch more churners early, reduce customer loss, and maximize retention impact with minimal additional effort.

APPENDIX

Happy Learning !