Business-Oriented Churn Prediction: Using ANN and XGBoost Tuning Thresholds for Practical Banking Use Cases

Umar Mahmood Siddiqui

Abstract. Customer churn significantly impacts retail banks' profitability, necessitating effective predictive modeling strategies. This study evaluates the financial implications of threshold tuning and model complexity in churn prediction models using a publicly available bank customer dataset comprising 10,000 records with approximately a 20.4% churn rate. Three modeling techniques—Simple Artificial Neural Network (ANN), Deep ANN, and XGBoost, were each trained under three distinct classification thresholds: standard (0.5), lowered (0.3), and lowered with class weighting to manage class imbalance. Threshold adjustment markedly influenced recall across models, however, improved recall typically reduced precision. A cost-based evaluation revealed significant differences in financial outcomes: the high-recall, weighted-threshold XGBoost generated a net profit of €37,950 by capturing most potential churners, outweighing the costs of increased false positives. Conversely, its precision-oriented model with a standard threshold resulted in a net loss (€8,925), underscoring the high financial penalty of missed churners, even though both models outperformed the loss that would occur if no predictive model was applied. The study emphasizes that pragmatic threshold calibration, rather than merely increasing model complexity, optimizes profitability in customer churn management scenarios where customer lifetime value substantially exceeds retention intervention costs.

1. Introduction

Customer churn or attrition is a well-recognized problem which refers to the state in which a customer stops using the products and services of a company (Zhang, 2023). Churned customers can lead to a significant loss in revenue and profitability for any business. In banking, it has been reported that customer churn is one of the top sources of revenue leakage for banks, around 10-15% loss in annual gross revenue (Karthikeyan et al., 2017). There is a loss in the direct revenue for banks from churned customers along with the high cost of acquiring new customers, which can be at least five times the cost of retaining an existing one (Liu et al., 2022). Effective churn prediction enables banks to proactively identify clients at risk of leaving and implement targeted retention strategies, thereby reducing potential losses (Suguna et al., 2025)

Previous literature related to customer churn prediction techniques have utilized simple regression, decision trees. More recently, machine learning models have also been applied with such prediction problems, most notable of these have been ensemble tree-based models and Neural Networks. We will explore these more in our overview of literature. A challenge faced by all such prediction problems is the issue of class imbalance, which is one the author also faces in this study. To mitigate the issue of class imbalance, it is widely acceptable in literature that synthetic methods to improve model accuracy are utilized. An example of one such method is SMOTE, however, for this study, the author focuses on how to reach the best possible outcome without the use of such techniques. This is done to make sure that the study is done with the least amount of pre-processing, a scenario that is common in practical business applications.

The dataset used in this study is an open-source dataset from Kaggle. This dataset contains some details about individual bank customers such as their bank balance, age and other relevant data. After minimal pre-processing, the author has experimented with different Machine Learning models to understand whether it is better to use models that are highly complex in order to solve the problem of class imbalance and achieve results that result in better sensitivity/ recall along with accuracy scores, or would it be more beneficial for businesses, especially banks, to prioritize business use case specific thresholds instead of more complex models.

This study enhances current literature by changing the focus from just looking at accuracy of models. The results of the study indicate that a more holistic view of business use cases should direct the use of machine learning models using threshold tuning. The idea is to assist business use case environments, where real-life scenarios require speed and efficiency of model implementation. This article explores the idea that utilizing deep neural networks can enhance predictive performance; however, the use of simpler, faster, and more explainable models, such as XGBoost can also provide businesses with a sufficiently effective alternative by solely focusing on threshold tuning. This method could be a more efficient method to solving churn problems and can be tweaked easily to different business use cases, resulting in better profitability.

2. Literature review

A rich body of literature has emerged on customer churn prediction techniques. Early approaches often relied on traditional statistical models like logistic regression or decision trees (Cao et al., 2024; De Caigny et al., 2018) . More sophisticated classifiers have now been applied with the rise of Machine Learning Models such as ensemble tree-based methods (e.g. Random Forests, Gradient Boosting) and neural networks. These have shown superior performance by capturing nonlinear patterns in customer behaviour (Brânduşoiu et al., 2024). In a study on bank customer churn, de Lima Lemos et al. (2022) applied numerous machine learning algorithms to large-scale banking data, and found that ensemble methods, especially Random Forest, are often superior as compared to simpler models in churn classification. Going further beyond to more complex model, recent developments have shown that deep learning techniques and hybrid models can outperform even Random Forests. Esmaeilpour Charandabi (2023) showed that a properly tuned shallow ANN (with a single hidden layer) outperformed Random Forest on a 10,000-customer banking dataset, while being robust to overfitting and noise. Advanced architectures, such as recurrent neural networks and attention-based models, have also been proposed to incorporate sequential customer data or to boost performance by small margins (Mena et al., 2023). However, these complex models often act as "black boxes," leading to challenges in interpretability (Guidotti et al., 2018)

A common challenge in banking churn modelling cases is the class imbalance between the majority group of retained customers as compared to the minority of customers who churn. In our dataset (which also reflects the real-world pattern), there are approximately 20% of customers churned, meaning a naive model predicting "no churn" for all would be correct 80% of the time. This would result in a high accuracy score but will not provide any meaningful help in predicting churners. Traditional accuracy-focused evaluation can thus be misleading (Suguna et al., 2025). Researchers

have noted that classifiers tuned for maximum accuracy tend to favor the majority class, yielding good accuracy but very low recall for the churn class (Suguna et al., 2025). As a result, there is growing recognition that precision, recall, and cost-oriented metrics are appropriate for churn problems than raw accuracy (AbdelAziz et al., 2025). Precision measures how many of the customers predicted to churn do so, this is important when intervention resources are limited or if the cost of reaching out to customers to prevent them from churning is high. Recall (sensitivity) measures how many of the actual churners the model manages to identify, this is crucial when the cost of missing a churner is high, which is usually the case, as the cost of acquiring a customer is usually high. The harmonic mean of these, which is the F1-score balances the two.

Burez and Van den Poel (2009) showed evidence from the telecom sector that AUC, lift, precision and recall outperform accuracy for rare-event churn because they do not "place more emphasis on one class over the other". Imani et al. (2025) conducted a survey of 240 studies and support a similar conclusion, flagging "profit-driven frameworks" and "business-oriented metrics" as a key emerging trend in churn modelling. They also demonstrate the value of cost-sensitive machine learning algorithms that directly incorporate misclassification costs into model training. Similarly, contemporary reviews stress the importance of profit-driven evaluation frameworks and the consideration of real-world deployment factors such as marketing costs and customer lifetime value (Imani, Joudaki, Beikmohamadi, & Arabnia, 2025)

3. Approach

Business-driven model: In practical terms, a bank's (or any business organization's) retention strategy might prioritize different metrics depending on context or cost structures. If the goal is an extensive retention-based campaign, where retaining customers is high priority, and therefore, contacting all likely customers with a special offer is required, *maximizing recall* is crucial as every missed churner is a lost opportunity. However, if sending these retention offers is expensive or resources are scarce, *maximizing precision* becomes important, as the bank would focus only on customers most likely to churn to avoid wasting effort on false alarms. A balanced approach might be used for general churn reduction efforts when both false positives and false negatives carry costs. Rather than treating churn prediction as a one-size-fits-all accuracy optimization problem, it is beneficial to tailor the model to these specific business objectives. This approach has become a goal in recent studies. For example, Höppner et al. (2020) introduced profit-based decision trees that directly maximize financial return of retention campaigns.

3.1 Study objective

Building on this evolving paradigm, our study aims to bridge the gap between technical model performance and business objectives in churn prediction. We investigate how adjusting **classification thresholds** and **model complexity** affects churn prediction outcomes in a banking scenario, and how these adjustments can be used to align with different business goals. Using a publicly available bank customer churn dataset, we train three models (a simple ANN, a deeper ANN, and one XGBoost model) under three different threshold or weighting settings, producing nine models in total. By evaluating each on precision, recall, F1 and accuracy, we identify which models are best suited for **high-recall (catch-all)** strategies, **balanced** strategies, or **high-precision**

(conservative) strategies. Furthermore, we perform a simple **profit analysis** by assigning monetary values to correct and incorrect predictions (true positives, false positives, etc.), inspired by approaches that "put a price" on churn (e.g. Blount, 2020). The profitability analysis was done based on concrete evidence for the true costs of our predictive modelling results and therefore, figures calculated by Schmitt et al., 2011 were utilized. The purpose of this analysis was to illustrate the real financial impact of each model, helping to determine which approach would indeed be most profitable or cost-effective for a bank. By taking this business-oriented perspective, our work contributes practical insights on how data science solutions can be tuned for deployment in real-world banking contexts.

4. Methodology and Data

4.1 Dataset and Preprocessing

We utilized a publicly available **Bank Customer Churn** dataset from Kaggle. The aim during this exercise was to reduce the steps required to process the data, as it would reflect business scenarios where quick procedures are preferred. This dataset contains 10,000 bank customers with various attributes and a binary label indicating whether each customer "exited" (i.e. churned) or not. The churn rate is approximately 20.4%, meaning 2,037 customers ended their relationship with the bank while 7,963 stayed. The features include a mix of demographic, account status, and activity variables commonly found in banking churn studies. Key features are:

- **Demographics:** Customer's *Age* (mean 39 years), *Gender* (roughly equal male/female distribution), and *Geography* (the branch/country of residence).
- **Financial status:** *Credit Score*, *Balance* (bank account balance), *Estimated Salary*, and number of bank products held (*NumOfProducts*).
- **Account activity:** Whether the customer has a credit card (*HasCrCard*), whether they are an active member (*IsActiveMember*), and their *Tenure* (years with the bank).

The dataset required minimal preprocessing. We removed a few identifier columns (Row Number, Customer ID, Surname) that have no predictive value. Categorical variables were processed via one-hot encoding: we converted *Gender* into binary (Male/Female) and *Geography* into dummy variables (the dataset includes countries like France, Spain, Germany). Continuous features were left in their original scale since the algorithms used (tree-based and neural nets with normalization in network layers) can handle them; however, we did check for outliers and distributions during exploratory analysis. No severe outliers were noted, and we applied feature scaling for all models. The data was split into training and testing subsets (80/20 split) to allow evaluation on unseen customers.

4.2 Modeling Techniques

We trained three different model types to represent a spectrum of model complexity and approach:

- **Simple ANN:** A feed-forward artificial neural network with two hidden layers (7 neurons each) and a sigmoid output neuron for binary classification. This network uses ReLU activation in hidden layers and was trained with the Adam optimizer (learning rate 0.001) for 50 epochs. We refer to this as the "Simple ANN" – it has a relatively small number of parameters and should be less prone to overfitting given the dataset size. Such a shallow network serves as a baseline for neural network performance

- **Deep ANN:** A deeper neural network with three hidden layers, having 16, 12, and 7 neurons respectively. Hidden layers also use ReLU, and output is sigmoid. This "Deep ANN" increases model complexity, potentially capturing more intricate nonlinear relationships at the cost of more parameters. It was trained under the same conditions (Adam optimizer, 50 epochs, batch size 32) as the simple ANN. The architecture choice (16-12-7) was somewhat heuristic but aimed to provide a larger capacity model that still trains efficiently on 10k samples. We monitored training loss to ensure the deeper network did not overfit; no early stopping was used, but 50 epochs was sufficient to converge in both ANN models.
- **XGBoost:** An implementation of the Extreme Gradient Boosting algorithm (tree-based ensemble) with 100 decision trees, maximum tree depth of 6, and a learning rate of 0.3. XGBoost is a powerful gradient boosting machine known for its strong performance on structured data. We used default regularization parameters and allowed the model to run until 100 trees as a reasonable trade-off between bias and variance. This model was used to check if there are similar results when using another model.

All models output a **probability of churn** for each customer. To convert these probabilities into a binary prediction (churn or not), a **classification threshold** must be applied. The default threshold is 0.5, meaning customers with predicted probability ≥ 0.5 are classified as churners. However, as discussed, we also consider alternative thresholds to bias predictions towards the positive (churn) class. One important point to consider here is that the model performances of ANNs change slightly whenever they are trained again, however, the results do not vary significantly each time they are run.

4.3 Threshold and Class Weight Configurations

We evaluated each model under three different configurations to simulate various business objective scenarios:

1. **Normal threshold (0.5), no class weight:** This is the baseline where the model is treated in the standard way, aiming roughly to maximize accuracy. A threshold of 0.5 usually balances precision and recall for a well-calibrated model. No class weighting means the training process did not artificially adjust for class imbalance, that is, the machine learning model saw the true distribution of churners vs non-churners. This setup is expected to yield high overall accuracy but potentially lower recall for churners, since the models might lean towards predicting the majority class (non-churn) to minimize error.
2. **Lowered threshold (0.3), no class weight:** Here we adjust the *decision threshold* to 0.3 while keeping training unchanged. By lowering the cutoff, we make the classifier more sensitive, more customers will be predicted as "churn" (positive) than at 0.5, which should increase recall (catch more of the actual churners) at the expense of precision (more false positives). This configuration is motivated by the scenario of prioritizing recall: if false positive interventions are not too costly, it may be better to identify as many potential churners as possible. Technically, this does not change the model's learned parameters; it only changes how we interpret the output probabilities. If the model's probabilities are well-calibrated, moving the threshold to 0.3 corresponds to treating a predicted 30% churn risk as the action trigger for retention efforts.

3. **Lowered threshold (0.3) with class weighting:** In this configuration, we introduce a class weight for churn during training *and* use the 0.3 threshold at prediction time. Class weighting means we inform the model that mistakes on the positive class (churn) are more costly than mistakes on the negative class. This effectively makes the training process more recall oriented as well, possibly resulting in a different decision boundary and internal model than an unweighted model. Combining this with the 0.3 threshold at inference further reinforces the bias towards predicting churn. This setup aligns with a highly aggressive retention strategy where the model is explicitly optimized to not miss churners. We expect this to yield the highest recall of the three, potentially with some additional false positives.

In total, we train 3 model types × 3 settings = **9 models**. All models are evaluated on the same test set to ensure comparability. It is worth noting that we did not use any oversampling (such as SMOTE) in this study, relying instead on class weighting as our imbalance mitigation strategy. This choice was to isolate the effect of threshold and weighting on the model itself. However, in other studies, SMOTE and ensemble combinations have been very effective, for instance, Suguna et al. (2025) achieved an F1-score of approximately 0.876 by applying SMOTE sampling alongside ensemble classifiers in churn prediction models.

4.4 Evaluation Metrics

We focus on four evaluation metrics – **Accuracy, Precision, Recall, and F1-score** – as they relate to different business goals:

- **Accuracy:** the proportion of all predictions that were correct. While commonly used, accuracy can be misleading for churn due to class imbalance. In our case, a model that predicts every customer will stay (no churn) gets approximately 80% accuracy by default. Thus, we use accuracy primarily for completeness, but not as the sole criterion.
- **Precision (Positive Predictive Value):** the fraction of predicted churners who churned. High precision means when the model flags a customer as likely to churn, it is usually correct. This metric is crucial if contacting a customer who wouldn't have churned is costly or undesirable (e.g., offering unnecessary incentives or upsetting a customer by suggesting they might leave). For example, if precision is 0.75, that means 75% of retention offers would go to actual at-risk customers, and 25% would be wasted on customers who would have stayed anyway. In some businesses, an offer or intervention cost could be higher, hence so precision might be more valued in those cases.
- **Recall (Sensitivity):** The fraction of actual churners that the model correctly identified. High recall means the model is catching most of the customers who will churn, which is vital if each lost customer has a large revenue impact. If recall is 0.85, the bank can intervene on 85% of those who would have churned (the other 15% slip through). A recall-oriented model helps ensure the bank's retention campaign reaches almost all truly at-risk customers.
- **F1-Score:** The harmonic means of precision and recall. It provides a single measure that balances the two, useful when we want a general assessment of model performance on the minority class.

  For each of the 9 models, we compute these metrics on the test set. This allows us to compare, for instance, the recall of the ANN with threshold 0.3 vs the recall of XGBoost with threshold 0.5, etc., to see how threshold and model type interact.

5. Results

Performance of Models at Different Thresholds

The classification results illustrate the trade-offs between precision and recall induced by threshold tuning and class weighting. Table 1 summarizes the accuracy, precision, recall, and F1-score for all nine model configurations.

For brevity, we highlight key comparisons and trends here (each model identifier is abbreviated as [Model]-[Threshold]-[Weight], where Threshold is H for 0.5, L for 0.3, and "+W" indicates class-weighted training):

- **Baseline models (0.5 threshold, no weighting):** All three models achieved high accuracy around 80–85%, reflecting that most non-churners were correctly identified (the baseline accuracy if predicting all non-churn would be 79.6%). However, as expected, their recall for churn was relatively modest. The Simple ANN (ANN-H) and Deep ANN (D-ANN-H) captured less than half of the churners, while XGBoost (XGB-H) captured half of the churners. These outcomes show that unadjusted models tend to prioritize overall accuracy and thus under-identify churners. Notably, the Deep ANN at this threshold setting slightly outperformed the Simple ANN on most metrics (e.g., a few points higher in recall and precision), suggesting the deeper architecture could learn more nuanced churn indicators. More complex models like deeper NNs or boosted trees can yield better churn prediction than simpler networks.

Table 1: Model results

Model identifier	Accuracy	Precision	Recall	F1-score	AUC
ANN_simple_threshold_05	0.861	0.814	0.408	0.543	0.857
ANN_simple_thresh03_no_weights	0.832	0.577	0.644	0.609	0.856
ANN_simple_threshold_03_weighted	0.587	0.314	0.875	0.463	0.829
ANN_deep_thresh05	0.863	0.824	0.415	0.552	0.849
ANN_deep_thresh03	0.854	0.659	0.580	0.617	0.858
ANN_deep_thresh03_weighted	0.653	0.357	0.885	0.509	0.861
XGBoost_thresh05	0.853	0.697	0.491	0.576	0.829
XGBoost_thresh03	0.825	0.562	0.627	0.592	0.829
XGBoost_thresh03_weighted	0.774	0.466	0.769	0.580	0.836

- **Lower threshold models (0.3, no weighting):** Lowering the decision threshold influenced recall for all models, validating our expectation. For example, the Deep ANN at 0.3 (D-ANN-L) achieved a recall of roughly **58%** of churners, up from around 41% at the 0.5 threshold. This means the model caught more churners. Similarly, XGBoost at 0.3 (XGB-L) saw recall jump to 63%, and the Simple ANN also improved to 64%. These high recalls came with a trade-off: precision dropped for each. The deep ANN's precision fell to around **66%** (meaning almost half of those it predicted to churn did *not* churn), and XGBoost's precision dropped to 56%. Overall accuracy of these models decreased because of the increased false positives, a

sacrifice of some accuracy to gain recall. This illustrates an important point: if the aim is to improve identification of churners, tuning the threshold is an effective lever.

- **Weighted + lower threshold models:** Adding class weighting during training in addition to a low threshold produced the most aggressive models in terms of capturing churn. The Deep ANN with weighting (D-ANN-L+W) achieved the highest recall of all, reaching **88%** of churners identified (out of every 10 churned customers, it caught almost 9). Precision for this model was naturally the lowest of the group with ANNs scoring below 40% each and XGBoost scoring slightly above 40%, since they over-predicted churn.

Our results confirm that **adjusting the decision threshold is an effective strategy to control the balance between precision and recall**. Without any change in the underlying model, we were able to significantly alter model behavior: high threshold yields a *precision-focused* model, low threshold yields a *recall-focused* model. Class weighting further nudges the models towards recall, essentially integrating a cost sensitivity into training. The outcomes echo the insights of cost-sensitive learning research in churn management (Imani et al., 2025), which argues that aligning the model's objective with business costs is crucial. Our contribution here is demonstrating this alignment in a simple, practitioner-friendly way (threshold tuning) and quantifying the results for a banking use case.

5.1 Profit Analysis

To translate model performance into business terms, we performed a **profit/loss analysis** using a simple cost model. We assigned monetary values to the outcomes of churn prediction as follows:

- **True Positive (TP)** – correctly predicting a churner (and presumably intervening to retain them): we assume the bank *saves* €250 of revenue that would have been lost (also referred to as the Customer Lifetime Value), minus a small intervention cost (e.g., offering a discount, making a retention call) of €25. Thus, net **gain per TP = €225**.
- **False Positive (FP)** – predicting churn for a customer who would not have churned: the bank will unnecessarily intervene. This incurs the intervention cost with no revenue benefit. We assume **cost per FP = €25** (the cost of the retention effort). The customer stays, but they might have stayed anyway; at least we haven't lost them, but we spent resources needlessly. In many cases, false positives are relatively inexpensive (a phone call, an email offer), which is why a strategy can afford some false positives if it catches more true churners.
- **False Negative (FN)** – failing to predict a churner (the customer leaves un-warned): this is the most expensive error. The bank loses the customer's future revenue. We count **cost per FN = €250 (the loss in Customer Lifetime Value)**, equal to the revenue we could have saved with a successful intervention. In our simplistic model, we don't assign an extra intervention cost here because no intervention was made (indeed, that's part of the problem).
- **True Negative (TN)** – correctly predicting a non-churner (no action taken, customer stays): **cost/gain per TN = €0**. There is no direct impact – we rightly left a happy customer alone.

Using these values, we calculate a **Profit (or Loss) = 225 * (Number of TP) – 25 * (FP) – 250 * (FN)** for the predictions of each model on the test set. This gives a monetary figure interpretation of how well each model's predictions would do for the bank compared to a baseline profit/ loss where no model was used. It complements the precision/recall metrics by answering: if this model were

deployed in a campaign, how much money would we expect to save or lose? It's important to note that our profit analysis assumes the bank can retain every customer that it correctly identifies as at risk (i.e., the intervention always works for true positives). Interventions might succeed only a fraction of the time, which would scale down the TP benefit. Our analysis therefore represents an optimistic upper bound of retention impact. Nonetheless, it is useful for comparing models under consistent assumptions.

The values assigned to our True Positive, True Negative, False Positive and False Negative have been sourced from a study conducted by Schmitt, Skiera, & Van den Bulte, 2011. More details regarding the sourcing of these values can be found in the appendix.

5.2 Profit Analysis Results

The ultimate measure of success for a churn prediction model in a business context is the **financial impact**: Using the TP/FP/FN cost assumptions described earlier, we calculated the expected profit (or loss) for each model if its predictions were used to drive a churn intervention campaign.

To check whether it is useful to deploy a churn prediction model, it is useful to compare the results of the profit and loss of the model compared to a baseline figure where no model is used. For this, we calculate the profit and loss using the results of our test set. Out of the 2000 datapoints, we find that the test set contains 1593 actual non churners and 407 actual churners. This means that our True Negatives are 1593 and False Negatives (churners that we never reach out to) are 407. Based on this, the total loss we encounter will be the loss of the CLV for 407 customers which gives us a **loss of € 101,750.** We will keep this figure for our analysis as the baseline.

We used the XGBoost model for our profit analysis, as using any model configuration will give the same general trend due to similar affects of threshold tuning and class weights. Using the ANN models would also result in similar trends. Therefore, it is suitable to compare the results that differ using the same model with different thresholds. The main reason why the XGBoost model is preferred in this profit analysis is because the scores for its accuracy, recall and precision do not change at every new training instance, unlike the ANN models. Therefore, XGBoost gives us replicable results for our study.

- **High-Recall Model Yields Positive Net Profit:** The model configuration that prioritized recall (XGBoost with threshold 0.3 and class weighting) ended up being profitable, resulting in a profit of €37,950 because of deploying the model with the given values of CLV on the test set. By catching most of the churners, this model retained a large portion of the potential revenue (TP benefit), and although it sent out many interventions (including to false positives), the relatively low cost of those interventions did not eat up the savings. In total, the intervention costs for false positives were far smaller than the revenue saved from true positives, leading to a positive balance.
- **Precision-Focused (High-Threshold) Model Loses Revenue:** On the other end, the model that was more precision-oriented (0.5 threshold, which had higher precision but lower recall) had an unfavourable outcome with a loss of €8,925 (although still better than the baseline

loss where no predictive model was used). It predicted churn for relatively few customers – mostly the surest churners – so it missed many churners. The false negatives (missed churners) incur a heavy cost. Even though it didn't waste much on false positives, the loss of many customers led to a net **negative**.

These profit results confirm the intuition that **the cost ratio of false negatives to false positives is the driving factor** in deciding the optimal model. In our scenario, FN was 5 times worse than FP; therefore, the model with the highest recall (fewest FN) was optimal, even though it had more FP.

To test a hypothetical example where FPs might be higher, our analysis showed that in such a case, the precision focused model resulted in better profitability as compared to the recall focused model. However, these were only hypothetical values used only to confirm this trend. The values for this hypothetical analysis for TP, TN, FP and FN were 150, 0, -250 and -400 respectively and they resulted in a baseline loss of €162,800. The precision focused model performed the best, even though it resulted in a loss of €74,550, while the recall focused model resulted in a loss of €80,400.

From a business perspective, this means that if the bank can afford the operational effort, they should choose the churn prediction model with high recall, as each retained customer holds significant value. However, it's important to consider scenarios where the assumptions might differ. For instance, if interventions were very expensive or if an overly broad campaign had intangible downsides (e.g., annoying customers with unnecessary calls could itself lead to dissatisfaction), then the cost of false positives would rise. Our framework can accommodate different costs: one could plug in a higher FP cost and re-compute profits to find the threshold at which profit is maximized.

In summary, under realistic assumptions for banking, our results strongly suggest that a **recall-optimized churn model yields the best financial outcome**. The differences in raw predictive metrics translate into very large differences in monetary impact. It also underlines that evaluating churn models requires looking beyond accuracy and even beyond precision.

6. Conclusion

In this study, we approached customer churn prediction from a business utility standpoint, using the example of a retail bank. We built and evaluated multiple models (ANNs and XGBoost) under varying decision thresholds and class weight settings to illustrate how the choice of threshold and model can be tuned to specific business objectives such as maximizing recall for customer retention or maximizing precision to conserve resources. Our study mainly focuses on how to most effectively and efficiently improve profitability for an organization such as a bank with minimal pre-processing and computational power to get to the desired result. We conclude that it may not be a necessity to use a slow, inefficient and inexplicable model, such as ANNs, for better churn prediction and reduce loss, but it may also be suitable to utilize a simpler, more efficient and explainable model, such as XGBoost, with lower thresholds to predict churners. Our experiments led to several key findings:

- **Threshold tuning is a powerful lever:** Simply lowering the classification threshold from 0.5 to 0.3 led to a significant jump in recall with an acceptable drop in precision. This demonstrates that if a business goal is to catch as many churners as possible, adjusting the

threshold of an existing model is an easy and effective step. Conversely, if precision is paramount, one could even consider higher thresholds than 0.5 (not explicitly done in our study, but conceptually straightforward) to only target the most likely churners. Our contribution was quantifying this in a banking context and linking it to dollars saved or lost.

- **Model complexity and type matter, but no one-size-fits-all:** When deploying churn models, practitioners should consider an array of models and configurations and pick not just the model with highest overall accuracy, but the model that meets the desired balance of precision/recall for the task at hand. In some cases, a slightly less accurate model may be preferable if it yields, say, much higher recall, thereby saving more customers. Moreover, the study also showcases that using a simpler model, such as an XGBoost model compared to an ANN, can also lead to adequate predictive performance with a much lower inference time, therefore, they might be more affective in business situations where prediction speed might be important. Simply lowering the threshold value can lead to similar, if not the same results with a much faster and easier to implement model.
- **Business-oriented evaluation is essential:** By translating confusion matrix outcomes into profit estimates, we made explicit that a model's value to the business cannot be judged by accuracy alone, it must be judged by costs and benefits. The operational implications are that the bank should be prepared to contact a sizable number of customers (with the understanding that many will not have churned on their own, but this precaution is taken to prevent those who would). Given the low unit cost of outreach (e.g., a phone call or email), this strategy is justified by the high value of retained customers. Our analysis showed a clear net positive return for this approach under reasonable assumptions. However, if circumstances change, e.g., if the cost per contact rises substantially or resources are constrained, the bank could re-evaluate the threshold. The framework we provided allows for recalculating profit under new cost assumptions easily.

7. Limitations and Future Work

Our study has three key limitations that suggest avenues for future improvements:

- *Lack of interpretability:* Neither neural networks nor boosted tree ensembles are easily interpretable out of the box. In a practical setting, bank officers might want to know **why** the model flags certain customers as churn risks (for example, is it because their balance dropped, or they've reduced product usage?). We did not incorporate an interpretability layer such as SHAP (SHapley Additive exPlanations) or LIME in our analysis.
- *Generality of results (single dataset):* We tested our approach on one dataset. While it is a realistic one, it may not capture all nuances of different banks (e.g., this data was somewhat limited in geographic scope and perhaps period). The churn behavior in other markets or in current times might involve additional features. Future work should validate the business-driven modeling approach on **multiple datasets**
- *Cost assumptions and retention effectiveness:* Our profit analysis was illustrative. In practice, the actual saved revenue per retained customer and the actual cost per contact could differ. Moreover, not every customer contacted will be retained – some will churn despite intervention, and some might have stayed anyway even without contact. Modeling these probabilities would make the profit analysis more robust.

Appendix

Profit analysis cost calculation derivation

Bottom-line numbers: We use well documented academic evidence available for European retail banking which states that a single mass-market customer is worth roughly €250 CLV and it costs €25 to send a retention incentive (the reward used in German bank referral/loyalty programmes). With those two inputs, the cost (or profit) associated with each cell of a churn-model confusion matrix is:

Outcome	Cash out-flow (-) / in-flow (+)	How it is calculated
TP – predicted to churn & really would	+ €225 net profit (€250 CLV saved – €25 retention cost)	We spend the €25 offer and avert the €250 loss, so we are €225 ahead.
FP – predicted to churn but would have stayed	– €25	Offer wasted; no CLV change.
FN – predicted to stay but churns	– €250	Customer leaves; full CLV is lost.
TN – predicted to stay & really stays	€0	No action, no loss.

The table below showcases how the values are sourced:

Parameter	Value	Evidence	Derivation
Customer lifetime value (CLV)	≈ €250 (non-referred, 6-year NPV)	Schmitt, Skiera & Van den Bulte follow 9,814 retail-bank customers for 33 months and compute six-year CLVs; the paper reports that referred customers are €40 (≈ 16 %) more valuable than non-referred ones. The base (non-referred) CLV therefore equals 40 / 0.16 ≈ €250. (Schmitt, Skiera, & Van den Bulte, 2011)	40 € ÷ 16 % = 250 €
Retention-offer cost	€25 per targeted customer	The same study (and German banking practice) pays €25 to the referrer for each successfully retained or acquired customer; the authors note that "most German banks offer 25 euros for a referral"	Direct company outlay

These numbers sit comfortably inside broader banking economics where research shows that acquiring a new client costs 5 times more than retaining one (Hart, Heskett, & Sasser, 1990; Reichheld, 1996)

Why these figures are defensible?

- Empirical depth – the Schmitt et al. panel contains "real money" contribution margins for nearly 10 k accounts over almost three years, projected out to six years for NPV
- External corroboration – other European banking CLV work (e.g., Haenlein et al.'s (2007) 6.2 million-account Markov model) confirms similar magnitudes, even though results are anonymized in "currency units" for confidentiality
- Cost-sensitive modelling literature – profit-driven metrics such as Expected Maximum Profit (EMP) explicitly use these per-cell costs when choosing optimal cut-offs, and churn-model accuracy studies stress that false negatives are typically an order of magnitude more expensive than false positives (Verbraken, T., Verbeke, W., & Baesens, B. (2013)

References:

1. Zhang, W. (2023, March 14). Bank customer churn analysis and prediction. In Proceedings of the 4th Management Science Informatization and Economic Innovation Development Conference (MSIEID 2022), December 9–11, 2022, Chongqing, China. EAI. https://doi.org/10.4108/eai.9-12-2022.2327608
2. Karthikeyan, S., Goyal, D., Khodabandeh, S., Dye, T., & Chhajer, S. (2017, July 5). How banks can close the back door on attrition. Boston Consulting Group. https://www.bcg.com/publications/2017/financial-institutions-marketing-sales-how-banks-close-back-door-attrition
3. Liu, Y., Shengdong, M., Jijian, G., & Nedjah, N. (2022). Intelligent prediction of customer churn with a fused attentional deep learning model. Mathematics, 10(24), Article 4733. https://doi.org/10.3390/math10244733
4. Suguna, R., Suriya Prakash, J., Pai, H. A., Mahesh, T. R., Kumar, V. V., & Yimer, T. E. (2025, May 9). Mitigating class imbalance in churn prediction with ensemble methods and SMOTE. Scientific Reports, 15, Article 16256. https://doi.org/10.1038/s41598-025-01031-0
5. Cao, G., Jia, H., & Qiu, Z. (2024). Customer Churn Prediction Based on Multiple Linear Regression and Random Forest. Applied and Computational Engineering. Retrieved from [ACE Journal].
6. Brândușoiu, I., et al. (2024). *Customer churn prediction with hybrid resampling and ensemble learning*.
7. de Lima Lemos, R. A., Silva, T. C., & Tabak, B. M. (2022). *Propension to customer churn in a financial institution: A machine learning approach*. *Neural Computing and Applications, 34*(1), 11751–11768. https://doi.org/10.1007/s00521-022-07067-x
8. Esmaeilpour Charandabi, S. (2023). Prediction of Customer Churn in Banking Industry [Preprint]. arXiv. Retrieved from https://arxiv.org/abs/2301.13099
9. Mena, G., Coussement, K., De Bock, K. W., De Caigny, A., & Lessmann, S. (2023). Exploiting time-varying RFM measures for customer churn prediction with deep neural networks. Annals of Operations Research. https://doi.org/10.1007/s10479-023-05259-9
10. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D., & Giannotti, F. (2018). A survey of methods for explaining black box models. IEEE Access, 6, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052
11. AbdelAziz, N. M., Bekheet, M., Salah, A., El-Saber, N., & AbdelMoneim, W. T. (2025). A comprehensive evaluation of machine-learning and deep-learning models for churn prediction. Information, 16(7), 537. https://doi.org/10.3390/info16070537
12. Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3), 4626–4636. https://doi.org/10.1016/j.eswa.2008.05.027
13. Imani, M., Joudaki, M., Beikmohamadi, A., & Arabnia, H. R. (2025). Customer Churn Prediction: A Review of Recent Advances, Trends, and Challenges in Conventional Machine Learning and Deep Learning [Preprint]. Preprints.org. Retrieved from https://doi.org/10.20944/preprints202503.1969.v2
14. Höppner, S., Stripling, E., Baesens, B., Broucke, S. vanden, & Verdonck, T. (2020). Profit-driven decision trees for churn prediction. European Journal of Operational Research, 284(3), 920–933. https://doi.org/10.1016/j.ejor.2018.11.072
15. Blount, J. (2020, November 2). Putting a price on customer churn. Medium. https://medium.com/@stephen.blount99/putting-a-price-on-customer-churn-38a184e530b8
16. Schmitt, P., Skiera, B., & Van den Bulte, C. (2011). Referral programs and customer value. Journal of Marketing, 75(1), 46–59. https://doi.org/10.1509/jm.75.1.46
17. Hart, C. W., Heskett, J. L., & Sasser, W. E. (1990). The profitable art of service recovery. Harvard Business Review, 68(4), 148–156.
18. Reichheld, F. F. (1996). The loyalty effect: The hidden force behind growth, profits, and lasting value. Harvard Business School Press.
19. Haenlein, M., Kaplan, A. M., & Beeser, A. J. (2007). A model to determine customer lifetime value in a retail banking context. European Management Journal, 25(3), 221–234. https://doi.org/10.1016/j.emj.2007.01.004
20. Verbraken, T., Verbeke, W., & Baesens, B. (2013). A novel profit maximizing metric for measuring classification performance of customer churn prediction models. IEEE Transactions on Knowledge and Data Engineering, 25(5), 961–973. https://doi.org/10.1109/TKDE.2012.50

Dataset can be accessed using this link:
https://www.kaggle.com/datasets/filippoo/deep-learning-az-ann