Degree project in Technology

First level, 15 credits

Customer Churn Prediction in a Retail Bank

Supervised Machine Learning using Comprehensive Customer Data

MAXIMUS OCCASUS, ERICA NORÉN

Customer Churn Prediction in a Retail Bank: Supervised Machine Learning using Comprehensive Customer Data

Kundbortfallsprediktion för en retailbank: Övervakad maskininlärningsimplementering med omfattande kunddata

M. Occasus, *Student in Industrial Engineering and Management, KTH*, E. Norén, *Student in Industrial Engineering and Management, KTH*

Abstract

Customer churn is a significant concern in retail banking, as customer losses can reduce future revenue and weaken long-term profitability. This thesis develops and evaluates supervised machine learning models for predicting customer churn within the securities segment of a Swedish retail bank, using both transaction and portfolio data.

Three models are developed and compared: Logistic Regression, XGBoost, and Balanced Random Forest. A tailored churn definition combining account deactivation and economic inactivity is proposed and feature engineering is implemented in order to take advantage of the properties of the panel data. Rolling out-of-time cross-validation is used throughout to prevent temporal data leakage.

All three models achieve strong out-of-time ROC-AUC scores above 0.90, with XGBoost performing best at 0.951, followed by Random Forest at 0.943 and Logistic Regression at 0.917. XGBoost also achieves the strongest performance in terms of Precision@k, indicating that it is particularly effective at ranking true churners among the customers with the highest risk.

The findings suggest that the model could support targeted retention strategies by identifying the highest-risk customers, while also indicating that churn prediction may have relevance beyond the studied segment for understanding customer loyalty across the bank's broader customer base.

Sammanfattning

Kundbortfall är en viktig fråga för retailbanker, eftersom förlorade kunder kan minska framtida intäkter och försvaga lönsamheten på längre sikt. I denna uppsats utvecklas och utvärderas övervakade maskininlärningsmodeller för att prediktera kundbortfall inom värdepapperssegmentet hos en svensk retailbank, baserat på både transaktionsdata och portföljdata.

Tre modeller utvecklas och jämförs: Logistic Regression, XGBoost och Balanced Random Forest. Studien föreslår en anpassad definition av kundbortfall som kombinerar avslutade konton med ekonomiskt motiverad inaktivitet. Vidare genomförs feature engineering för att utnyttja egenskaperna hos datamaterialets panelstruktur. Genomgående används rullande out-of-time korsvalidering för att minska risken för temporalt dataläckage.

Samtliga tre modeller uppnår starka out of time ROC-AUC värden över 0,90. XGBoost presterar starkast med ett ROC-AUC värde på 0,951, följt av Random Forest på 0,943 och logistisk regression på 0,917. XGBoost uppnår även starkast resultat sett till Precision@k, vilket tyder på att modellen är särskilt effektiv på att rangordna faktiskt kundbortfall bland de kunder som bedöms ha högst risk.

Resultaten visar att modellen kan stödja riktade kundbevarande strategier genom att identifiera kunder som löper störst risk att lämna segmentet. Samtidigt indikerar resultaten att prediktion av kundbortfall kan vara relevant även bortom det studerade segmentet, genom att bidra till en bredare förståelse av kundlojalitet inom bankens bredare kundbas.

I. Introduction

ON behalf of a Swedish retail bank, this thesis aims to develop machine learning models for estimating churn risk among customers within the bank's savings and securities account segment. Customer churn is a central concept in customer analytics as it directly affects aspects like profitability, customer lifetime value, and strategic decision-making. The objective of this thesis is to develop a churn prediction model, using supervised machine learning methods.

The analysis is based on comprehensive customer data, provided by the bank, including both behavioral and financial features. Beyond evaluating prediction accuracy for the specific models and their results, the thesis will investigate which categories of customer data contribute to customer churn.

Customer churn is directly linked to a bank's future revenues and overall performance. Haenlein, Kaplan, and Beeser [1] show that customer lifetime value models can be used to estimate customer value in a retail banking context. Churn prediction can therefore also support broader analytical applications, including customer lifetime value estimation, where churn and retention risk are central components in assessing expected future customer profitability. Such insights can support customer segmentation, resource allocation, and retention strategies.

Measuring churn probability allows the bank to identify customers at risk of churn, implement proactive measures, and thereby reduce churn. In addition, analyzing the features that drive customer churn provides insight into customer behavior. These insights can support the development of the bank's products, its pricing strategy, and the segmentation of customers.

The thesis aims to investigate the probability of churn for customers in the bank's *Savings* segment. Existing churn prediction studies in banking have mainly focused on broader retail banking relationships. In contrast, churn in the securities account segment has received less attention, although Ahn, Kim, and Lee [2] show that the customer transaction patterns can be used to predict attrition in the securities industry. This thesis therefore contributes by studying churn in a savings and securities account context within a Swedish retail bank. Furthermore, the qualitative aim of the thesis is to discuss the benefits and challenges of using a churn model in the securities account segment.

II. Research Questions

The research questions of this thesis are the following:

- Can machine learning methods be used to develop a functional customer churn model for securities account holders within the bank?
- How do different machine learning methods perform in predicting customer churn?
- Which are the benefits and challenges of using a churn model to predict customer churn in the securities account segment?

III. Literature Review

Customer churn is a widely studied research topic, and churn modeling is an established practice in both banks and other companies. One thesis by Barr and Pettersson [3] emphasizes the importance for profitability of retaining existing customers rather than acquiring new ones. Hence, preventing customer churn becomes a central part of maximizing the total number of customers, and predicting churn enables the bank to identify at-risk customers before they leave the service.
Brito et al. [4] researched customer churn prediction in retail banking using machine learning methods, where they proposed a framework to model customer churn behavior in the context of highly imbalanced classes using undersampling from the majority class when using XGBoost and elastic net to identify patterns. They further enhanced the importance of data preprocessing for model performance. Unlike the present study, Brito et al. had demographical variables such as marital status and education level available for their models. Only one of such features (overall income) was however deemed top 10 in importance for their churn prediction models.

In another study by Njaya and Sithole [5] conducting a comparative analysis of the determinants of customer churn behavior in various industries in Asia, Latin America and Sub-Saharan Africa, a key finding was that marketers in all three regions could improve profits by 25–125% by reducing churn by as little as 5%.

Another study by Liu and Wu [6] further notes the importance of offering customers a comprehensive "one-stop shopping experience" (i.e., being able to find all relevant products at one bank) for customer retention. This conclusion makes the segment-specific perspective in this analysis especially relevant, since increased customer churn rates in one segment are at risk of spreading to other segments, thus lowering the profits of the bank as a whole.

Conditions on the Scandinavian securities markets also suggest that the speed of securities re-allocation is increasing quickly as smaller niche banks such as Avanza and Nordnet are becoming increasingly popular for this segment especially. [7] This contrasts with the findings of Liu and Wu [6], as the retail banking market seems to be dynamically adapting to the current surge in fintech.

A. Defining Customer Churn

Defining the churn itself is an important step in the development of a churn model and can be done in multiple different ways. In a thesis by J. Karlberg and M. Axén [8], the difficulties in defining churn for non-contractual and non-subscription-based services are discussed. The authors describe the need for a trade-off between wanting to define churn as early as possible while avoiding defining churn when the customer was only in an inactive phase.

Their final churn definition was created using two different equations consisting of four variables in total: *Longevity*, *Recency*, *Active days* and *Wavelength*. *Longevity* is defined as the number of days since the customer's first order, *Recency* as the number of days since the most recent order, and *Active Days* as the number of days on which the customer placed at least one order. The Wavelength variable was calculated as the average time between orders:

$$\text{Wavelength} = \frac{\text{Longevity} - \text{Recency}}{\text{Active days}}$$

With the churn factor calculated as:

$$\text{Churn factor} = \frac{\text{Days since last order}}{\text{Wavelength}}$$

A similar way to define churn will be used in this thesis due to the need to take customers' trading patterns into account when defining churn.

In a study by Yang and Chiu [9] investigating customer churn within the telecom service industry, customer churn can fall into one of several categories:

- *Involuntary churn* occurs when the provider of a subscription service has to terminate the service due to customers failing to make payments.
- *Unavoidable churn* is when the customer is permanently removed from the market, e.g. passes away.
- *Voluntary churn* is when the customer voluntarily terminates the service, possibly for another provider with a better offer.

The study by Yang and Chiu [9] concludes that customer retention is one of the most important issues in customer relationship management and further emphasizes the importance of data mining.

IV. Theory

A. Theory on Machine Learning Methods Used

a) Logistic Regression: Logistic regression will act as a baseline model due to its interpretability and its established role in risk modeling within the financial industry. With interpretable coefficients, it allows for a clearer presentation of results to those who may be less familiar with similar models.

Hastie et al. [10] describe Logistic Regression as modeling the probability of a binary outcome $Y \in \{0, 1\}$ conditional on predictors X. The model specifies:

$$\Pr(Y = 1 \mid X = x) = \frac{\exp(\beta_0 + x^T\beta)}{1 + \exp(\beta_0 + x^T\beta)}$$

which ensures that the predicted probabilities lie in $[0, 1]$. Equivalently, the log-odds (logit) are linear in the predictors:

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + x^T\beta$$

The coefficient β_j can therefore be interpreted as the change in log-odds of the outcome per unit increase in x_j, holding all other predictors constant. Parameters are estimated by maximizing the log-likelihood:

$$\ell(\beta) = \sum_{i=1}^{N} [y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))]$$

b) Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions. It offers high predictive accuracy, robustness to noise, and makes few distributional assumptions about the data, as it is a non-parametric method.

Hastie et al. [10] describe the algorithm as follows. Let $T_b(x)$ denote the prediction of the b-th tree trained on a bootstrap sample of the training data. For regression, the ensemble prediction is the average across all B trees:

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

For classification, the predicted class is determined by majority vote:

$$\hat{G}(x) = \text{argmax}_k \sum_{b=1}^{B} I(T_b(x) = k)$$

At each node split, only a random subset of $m \leq p$ predictors is considered, where p is the total number of features. A common default is $m = \sqrt{p}$ for classification. This randomization reduces the correlation between trees and improves generalization compared to a single decision tree or bagging alone.

c) Gradient Boosting (XGBoost): Gradient boosting also applies decision trees, but builds them sequentially using boosting, where each new tree corrects the residual errors of the previous ones. This allows for the same benefits as Random Forest while often achieving superior predictive accuracy, especially on larger tabular datasets.

Chen and Guestrin [11] describe XGBoost as a tree ensemble model where the prediction for observation i is the sum of K regression trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Model training minimizes a regularized objective function:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $l(y_i, \hat{y}_i)$ is a differentiable loss function measuring prediction error and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ penalizes tree complexity through the number of leaves T and the magnitude of leaf weights w. This regularization reduces overfitting compared to standard gradient boosting.

Trees are added sequentially. At each iteration t, the objective is approximated using a second-order Taylor expansion:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first- and second-order gradient statistics of the loss function. The use of second-order information allows XGBoost to take more precise update steps than standard gradient boosting, which relies only on first-order gradients.

B. LASSO for regularized regression

LASSO (Least Absolute Shrinkage and Selection Operator) performs feature selection by adding an L_1 penalty to the Logistic Regression objective, which encourages sparse solutions by shrinking some coefficient estimates to exactly zero. Friedman, Hastie, and Tibshirani [12] describe this approach for regularized generalized linear models, including two-class

Logistic Regression. The LASSO Logistic Regression minimizes:

$$\min_{\beta} \left[-\ell(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

where $\ell(\beta)$ is the log-likelihood of the Logistic Regression model and $\lambda \geq 0$ is a regularization parameter controlling the degree of shrinkage [10]. Features with coefficients shrunk to zero are effectively excluded from the model, making LASSO a natural tool for simultaneous estimation and feature selection.

C. VIF for Multicollinearity Diagnostics

The Variance Inflation Factor (VIF) is used to identify multicollinearity. For feature j, it is defined as:

$$VIF_j = \frac{1}{1 - R_j^2}$$

where R_j^2 is the coefficient of determination obtained by regressing feature j on all other features [10]. A value of $VIF_j = 1$ indicates no multicollinearity, while values exceeding 10 are commonly used as a threshold for problematic multicollinearity.

D. Evaluation Measures

a) Recall: measures the proportion of actual churn events correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where TP is the number of true positives and FN the number of false negatives.

b) Precision: measures the proportion of predicted churn events that are actual churn events:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where FP is the number of false positives.

c) ROC-AUC (Receiver Operating Curve - Area Under Curve): is the area under the receiver operating characteristic curve, plotting true positive rate against false positive rate across all classification thresholds:

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(t)\, d\,\text{FPR}(t)$$

A value of 1 indicates perfect discrimination, while 0.5 corresponds to random guessing.

d) Log Loss: Log Loss measures the quality of predicted probabilities by penalizing confident but incorrect predictions. For a binary classifier it is defined as:

$$\text{Log Loss} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

where p_i is the predicted probability of churn and $y_i \in \{0, 1\}$ is the true label [10]. Lower values indicate better probabilistic predictions. Unlike AUC, Log Loss is sensitive to the actual calibration of predicted probabilities and not only to the ranking of observations, making it a useful complement to ROC-AUC when evaluating models on imbalanced datasets.

e) Precision@k: A key evaluation metric used during feature selection was Precision@k, which answers the following question: among the k customers flagged as highest churn risk in a given month, how many actually churn? Formally, for each month t:

$$Precision@k_t = \frac{\text{True churners in top } k}{k}$$

The average across all evaluation months is then:

$$\overline{Precision@k} = \frac{1}{T} \sum_{t=1}^{T} Precision@k_t$$

This metric is particularly relevant for the practical use case of this thesis, where the bank would act on a fixed number of high-risk customers each month rather than applying a fixed classification threshold.

E. Challenges of Panel Data in Churn Modeling

Panel data for machine learning presents both advantages and challenges. On one hand, it enables the use of time-dependent variables to better capture behaviors and patterns that can only be observed over time. On the other hand, the repeated observations of the same individuals introduce temporal dependence and potential data leakage, which require more careful validation than in standard cross-sectional data.

1) Time-Based Splits: In Cerqua et al. [13], the standard approach to supervised machine learning splitting involves randomly selecting observations for a training set (often 80%) and a test set (often 20%). The model performance on out-of-sample unseen data works as a reliable way to measure model quality thanks to the assumption that there are no cross-sectional dependencies between the two sets. With panel data however, this assumption is immediately violated and the standard approach could lead to both cross-sectional and temporal data leakage. This would then lead to data from the test set leaking into the training set, meaning that the model would be trained on data from the test set and thereby essentially making the model evaluation unreliable. Rolling out-of-time cross-validation, as used in this thesis and described in Section V-E, addresses this issue.

2) Time-Dependency Within Individuals: When constructing and testing the models, each observation was treated as independent, meaning that an observation belonging to a customer in one month was assumed to be independent from the same customer's observations in adjacent months. This assumption is violated in practice, where customer-specific patterns are likely to influence churn probability over time. Treating each monthly observation as independent likely results in underestimated standard errors and thereby somewhat overstated confidence in model performance. The prediction task itself, however, does not depend on unbiased variance estimation, as the objective is to rank customers by churn risk within a given month rather than to perform statistical inference on model parameters. Explicitly modeling the causal relationship between variables and churn rate falls beyond the scope of this thesis, and the rolling out-of-time validation ensures that no future information from the same individual leaks into the training set.

F. Challenges of Class Imbalance

The data set used for this thesis exhibits high class imbalance, as customers determined as churned represent a small minority of the total observations. He and Cheng [14] address this issue, noting that binary classifiers used on imbalanced data sets tend to be biased toward the majority class (in this case non-churners). This often results in high overall accuracy, since it is possible for the classifier to classify all observations as non-event and still achieve a high total share of correct classifications. To deal with this issue when constructing the models, algorithm-specific solutions had to be implemented.

1) Class Imbalance in Logistic Regression: The findings of He and Cheng suggest that the most challenging feature of class imbalance is not the small number of observations in the rare class relative to the majority class, but rather the absolute rarity of the event class. This issue is especially prominent for Logistic Regression as it inherently has a "small sample bias" issue as a result of maximum likelihood being used for the estimation.

2) Class Imbalance in Random Forest: Since the Random Forest as proposed by Breiman [15] works by creating bootstrapped decision trees while being constructed to minimize the overall error rate, it will often by design focus on improving accuracy for the majority class. This will result in low accuracy for the minority class, which is undesirable when predicting customer churn. Chen et al. [16] suggest Balanced Random Forest (BRF) as a solution to the issue with class imbalance. This approach addresses the significant probability that there are few, if any, minority class observations in a randomly selected bootstrap sample. For each tree $b = 1, \ldots, B$:

1) Draw a bootstrap sample S_b^+ from the minority class with replacement, $|S_b^+| = n^+$
2) Draw a random sample S_b^- from the majority class with replacement, $|S_b^-| = n^+$
3) Train a decision tree h_b on the balanced sample $S_b = S_b^+ \cup S_b^-$

The final prediction is determined by majority vote across all trees:

$$\hat{y} = \arg\max_c \sum_{b=1}^{B} \mathbf{1}[h_b(x) = c]$$

where $h_b(x)$ denotes the prediction of tree b. Each bootstrap sample contains $2n^+$ observations, giving the minority class equal representation regardless of its true frequency in the data.

V. Methodology

A. Data

The data used for the analysis consisted of monthly snapshots of data from an internal back-office system containing customer-level information such as holdings across different securities account types and asset classes (stocks, funds and cash balance). The initial data set used contained around 5 million observations. An additional data set describing the time of the customer's first opened securities account within the bank, as well as, for some customers, the time of when the account was closed, was used to calculate features such as account age and tenure. This set featured the same customers as the one containing the financial variables, with some exceptions being made due to customers deactivating before the first month in the dataset.

B. Structure of the Raw Variables

1) Level Variables: Level variables include: *Total MV*, the total market value of all securities within the account excluding the cash balance; *Total MV and Cash*, the sum of the total market value and the cash balance; and *Total Cash*, the cash balance in the account. All are numerical variables.

2) Securities Account Types: Securities account types describe the type of account the customer has holdings in. The data set includes binary variables (1 for having holdings in the specific account type and 0 if not), and numerical variables for the cash balance in the securities account. The different types of securities accounts present in the data were ISK, VPD as well as ISKR, VPR and CIM whose content consists of equities that have been acquired using the advisor service offered by the bank.

3) Holding Types: The holding types specified in the columns are stocks and funds.

4) NCCF Columns: NCCF columns (Net Cumulative Cash Flow) are a set of variables describing customer activity, i.e. transactions and relocations of instruments. These were numerical and described the amount affected by the action, divided into the securities account and holding categories described above. These variables were especially relevant for the inactive churn definition implemented for this thesis since they are the strongest indicator of customer activity.

C. Data Preprocessing

The quality of the data received for the thesis was high throughout, and the only processing needed was analysis-specific procedures such as data type conversions in order to make the variables suit the structure of the modeling-process. In order to achieve optimal performance and to determine a suitable definition of customer churn, a number of observations were also removed.

1) Removing Irrelevant Early Observations: Early observations are retained only if the customer is economically relevant or transactionally active. This was done to minimize size of the data frame, and to not train the model on irrelevant observations. To ensure this, only customers who had activity within the time window of data were included:

$$MeaningfulActive_{i,t} = \begin{cases} 1 & \text{if } NCCF_{i,t} > 0 \\ 1 & \text{if } TotalMVCash_{i,t} \geq 500 \\ 0 & \text{otherwise} \end{cases}$$

The first meaningful month is defined as $t_i^{start} = \min\{t : MeaningfulActive_{i,t} = 1\}$. All earlier observations are removed.

D. Churn Definition

Defining churn can, as mentioned in the literature review, be challenging, and a tailored definition suitable for the data available had to be implemented. For the savings account

product especially, customer activity had to be taken into account since a customer could theoretically not be using the product actively and thereby not creating any value for the bank despite having holdings there. To deal with this challenge, two different churn definitions had to be used for defining the positive class.

1) Hard Churn (Deactivation): The most intuitive churn definition is customer IDs observed in the data set one month but not the following — explicitly meaning that they have actively deactivated their accounts and thereby left as customers. A *Deactivated* variable was thereby created, where $Deactivated_{i,t} = 1$ if customer i is observed at t but not at $t+1$, otherwise 0. *An exception was made for the last month in the dataset as there was no succeeding month available.*

2) Inactivity Churn: Inactivity churn had to be defined in a way that did not include an excessive amount of low-frequent customers in the positive class, while still taking the actual value created for the Bank into account. It is built up from the following components:

- *Transaction Activity.* Let $k \in K$ denote all transaction types in the NCCF data. Since transactions can be both positive and negative, total activity is defined as absolute transaction magnitude:
$$NCCF_{i,t} = \sum_{k \in K} |NCCF_{i,t,k}|$$
- *Monthly Inactivity.* A customer is considered inactive if no transaction activity occurs during month t:
$$Inactive_{i,t} = \begin{cases} 1 & \text{if } NCCF_{i,t} = 0 \\ 0 & \text{otherwise} \end{cases}$$
- *Six-Month Inactivity.* A customer must be inactive during the current month and the following five months, i.e. $Inactive_{i,\tau} = 1$ for all $\tau \in \{t, t+1, \ldots, t+5\}$:
$$Inactive6M_{i,t} = \begin{cases} 1 & \text{if } \sum_{\tau=t}^{t+5} Inactive_{i,\tau} = 6 \\ 0 & \text{otherwise} \end{cases}$$
If full future information does not exist, $Inactive6M_{i,t} =$ NaN.
- *Economic Churn.* A customer is considered to have churned economically if portfolio value is below SEK 500 and the customer has been inactive for six months:
$$EconomicChurn_{i,t} = \begin{cases} 1 & \text{if } TotalMVCash_{i,t} < 500 \\ & \quad \text{and } Inactive6M_{i,t} = 1 \\ 0 & \text{otherwise} \end{cases}$$

3) Final Churn Definition: A churn event occurs if either deactivation or economic churn is observed:
$$ChurnFlag_{i,t} = \begin{cases} 1 & \text{if } Deactivated_{i,t} = 1 \\ & \quad \text{or } EconomicChurn_{i,t} = 1 \\ 0 & \text{otherwise} \end{cases}$$

4) Prediction Target: The model predicts churn one month ahead:
$$Y_{i,t} = ChurnFlag_{i,t+1}$$
This corresponds to the variable `churn_t_plus_1`.

5) Right Censoring: Since $Inactive6M_{i,t}$ requires future data up to $t+5$, and the prediction target requires $t+1$, the total future data requirement is $t+6$. Given that the dataset ends at $T =$ *The final month of the dataset*, valid observations must satisfy:
$$t + 6 \leq T$$

E. Cross-Validation

To address the risk of temporal data leakage described in Section IV-E1, rolling out-of-time cross-validation as described by Cerqua et al. [13] is used (referred to as out-of-sample in their work). The data is partitioned into sequential folds, where each fold serves as the validation set exactly once while all preceding observations are used for training. The final performance estimate is the mean across all folds.

Training set at fold t:
$$Train_t = \{1, \ldots, t\}$$
Test set at fold t:
$$Test_t = \{t+1\}$$

F. Feature Engineering

Raw variables were transformed into engineered features capturing portfolio allocation, rolling market value and liquidity patterns, and changes in customer activity over time. The final feature set, including both raw variables and engineered features, contained nearly 200 features. Table I summarizes the engineered features.

TABLE I: Engineered Features

Feature	Definition
$ShareCash_{i,t}$	$Cash_{i,t}/TotalMVCash_{i,t}$
$ShareSec_{i,t}$	$MV_{i,t}/TotalMVCash_{i,t}$
$ShareFunds_{i,t}$	$FundMV_{i,t}/MV_{i,t}$
$ShareEquity_{i,t}$	$EquityMV_{i,t}/MV_{i,t}$
$MVMean_{i,t}^{(w)}$	$\frac{1}{w}\sum_{j=t-w+1}^{t} MV_{i,j}$, $w \in \{3, 6, 12\}$
$MVDelta_{i,t}^{(w)}$	$MV_{i,t} - MV_{i,t-w}$
$CashMean_{i,t}^{(w)}$	$\frac{1}{w}\sum_{j=t-w+1}^{t} Cash_{i,j}$
$CashDelta_{i,t}^{(w)}$	$Cash_{i,t} - Cash_{i,t-w}$
$NetInflow_{i,t}$	$Buy + Inflow + Sell + Outflow$
$MonthsSinceActivity_{i,t}$	$t - t_i^{lastactive}$
$Wavelength_{i,t}$	$(MonthsObs - MonthsSinceAct)/ActiveMonths$
$ActivityRatio_{i,t}$	$MonthsSinceActivity/Wavelength$

1) Handling Missing Values from Rolling Features: Rolling features such as $MVMean_{i,t}^{(w)}$ require w consecutive prior observations to be computed. For a customer i first observed at t_i^{start}, the rolling mean is undefined for all $t < t_i^{start} + w - 1$:
$$MVMean_{i,t}^{(w)} = \begin{cases} \frac{1}{w}\sum_{j=t-w+1}^{t} MV_{i,j} & \text{if } t \geq t_i^{start} + w - 1 \\ \text{NaN} & \text{otherwise} \end{cases}$$

These missing values cannot be imputed using future data, as this would introduce data leakage; the model would have access to information that is not yet available at the time of prediction. Similarly, backfilling with the first available rolling

value would allow information from period $t_i^{start} + w - 1$ to influence predictions at earlier time points.

The resulting NaN observations were therefore dropped from the training set. Since the longest rolling window used was $w = 12$ months, this means that at most the first 11 observations per customer are excluded. This represents a trade-off between retaining early observations and ensuring that all features are computed without future information. Given the timespan of the dataset, the majority of each customer's history is preserved.

G. Dealing with Imbalanced Classes

1) Class Weights for Logistic Regression: Class weights were introduced using scikit-learn's built-in *balanced* option [17]. The class weight for class j is computed as:

$$w_j = \frac{n}{k \cdot n_j}$$

where n is the total number of samples, k is the number of classes, and n_j is the number of samples in class j.

2) Balanced Random Forest: The Balanced Random Forest (IV-F2) was implemented using the *BalancedRandomForestClassifier* from the Python *Imbalanced Learning* library [18].

3) No Class Imbalance Adjustment for XGBoost: For XGBoost, model performance was noted to be better by a wide margin when class imbalance was not taken into account when choosing hyperparameters. The reason for this is evaluated in the Discussion section.

H. Feature Selection

The initial feature set was extensive and therefore required systematic reduction to limit noise, reduce multicollinearity, and mitigate the risk of overfitting. This process included a number of general and model-specific steps but also intuitive reasoning regarding business relevance.

1) General Step 1: Feature Family Grouping: Before any statistical filtering was performed, all engineered variables were manually grouped into economically meaningful feature families to avoid comparing fundamentally different types of variables directly against one another. The families included aggregated portfolio size features, lifecycle and activity features, portfolio allocation features, rolling market value and liquidity features, and transaction behavior features. Highly granular transaction variables such as account-specific transaction types were excluded early due to excessive dimensionality and low interpretability, reducing the initial candidate set within each family.

2) General Step 2: Correlation Screening: Within each feature family, pairwise correlations were analyzed to identify highly redundant variables. Variables were candidates for removal when their absolute correlation was high, when two variables captured nearly identical information, or when one variable was clearly more interpretable or economically meaningful than the other. Examples of redundant structures included total portfolio value versus highly overlapping subcomponents, binary activity indicators versus continuous transaction magnitudes, and short- and long-window versions of nearly identical rolling metrics.

Importantly, high correlation alone was not sufficient for removal. If two correlated variables captured conceptually different mechanisms — for example portfolio level versus portfolio change — both were retained for further evaluation.

After these two general steps, the feature selection process diverged depending on the model.

3) Logistic Regression Feature Selection:

a) Step 1: Univariate Predictive Screening: After the general steps, each remaining candidate feature was evaluated individually using rolling out-of-time Logistic Regression. For each feature, a model was trained using only that single variable, with expanding rolling time windows to avoid temporal leakage, and performance was evaluated on future unseen months. The evaluation criteria were ROC-AUC, standard deviation across folds, and missing value share. Features were removed when they performed close to random classification ($AUC \approx 0.5$), consistently underperformed stronger variables within the same family, or added little economic intuition.

b) Step 2: LASSO and VIF: Because Logistic Regression is sensitive to multicollinearity, LASSO (IV-B) as an additional selection step was performed. A LASSO Logistic Regression was trained on multiple random subsamples of the dataset and features were evaluated based on selection frequency across samples, coefficient stability, and coefficient magnitude. This allowed unstable variables to be removed while retaining consistently important predictors.

A final multicollinearity check was performed using the Variance Inflation Factor (VIF) (IV-C).

Features with extremely high VIF values were reviewed manually and removed when they represented nearly identical information. This produced a final Logistic Regression feature set with improved interpretability and coefficient stability.

4) XGBoost Feature Selection: Tree-based models are substantially more robust to multicollinearity and non-linear feature interactions than Logistic Regression. The univariate screening step was therefore not applied to XGBoost. Instead, after the general steps, feature importance was evaluated directly using both XGBoost gain importance and SHAP values, and features with consistently low importance across both methods were removed.

a) Step 1: Importance-Based Selection: Three candidate models were constructed using the top 15, 10 and 5 features respectively. In cases where two variables showed similar importance, preference was given to the one with lower missing value share or higher interpretability, for example retaining one rolling window length when two showed near-identical predictive power. The three models were then compared using out-of-time evaluation, with results shown in Table II.

TABLE II: XGBoost Feature Selection: Performance Comparison

Metric	Top 15	Top 10	Top 5
Precision@50	0.93	0.93	0.87
Precision@100	0.90	0.91	0.84
ROC-AUC	0.95	0.95	0.95
Log Loss	0.01	0.01	0.01

5) Random Forest Feature Selection: As with XGBoost, the univariate screening step was not applied to Random Forest given its robustness to correlated predictors and non-linear feature interactions. The same candidate set produced after the two general steps was retained, and feature importance scores were evaluated across rolling validation folds.

a) Step 1: Importance-Based Selection: Variables with consistently negligible importance were removed. Three candidate models using the top 15, 10 and 5 features were then constructed and compared, with results shown in Table III.

TABLE III: Random Forest Feature Selection: Performance Comparison

Metric	Top 15	Top 10	Top 5
Precision@50	0.90	0.90	0.89
Precision@100	0.86	0.87	0.86
ROC-AUC	0.94	0.94	0.89
Log Loss	0.04	0.04	0.08

6) Summary: The final feature selection framework was model-specific. Logistic Regression received the most aggressive filtering including univariate screening, LASSO, and VIF due to its sensitivity to multicollinearity, resulting in a compact and interpretable feature set. XGBoost and Random Forest bypassed the univariate screening step entirely, retaining broader candidate sets with final selection guided by importance scores. This ensured that each model received a feature set aligned with its methodological strengths while reducing unnecessary complexity and overfitting risk.

I. Hyperparameter tuning

No systematic hyperparameter optimization such as grid search was performed due to its computational costs, especially in combination with the rolling out-of-time cross validation used. Prior research such as Brito et al. [4] also suggests that feature engineering and selection tend to have a larger impact on churn model performance than hyperparameter tuning, making the latter a lower priority in the modeling pipeline. This was then confirmed to be the case for the model in this thesis too when evaluating a small sample of hyperparameter combinations.

VI. Results

A. Model Evaluation Figures

Figures 1, 2, and 3 show the out-of-time ROC curves for the three different models.

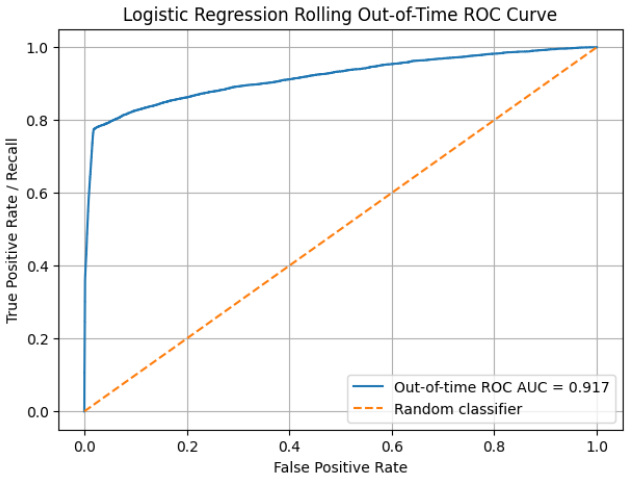

Fig. 1: ROC curve for the Logistic Regression model.

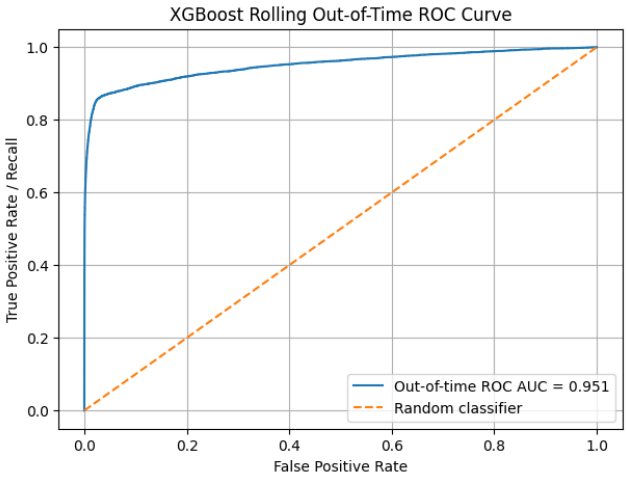

Fig. 2: ROC curve for the XGBoost model.

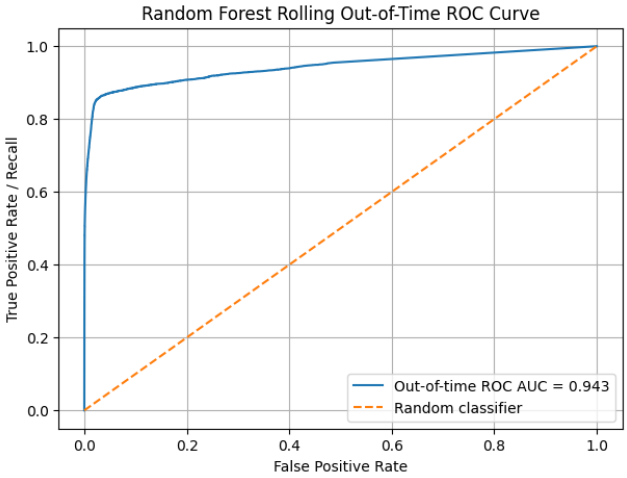

Fig. 3: ROC curve for the Random Forest model.

Figures 4, 5, and 6 show how precision changes across different classification thresholds for the three different models.

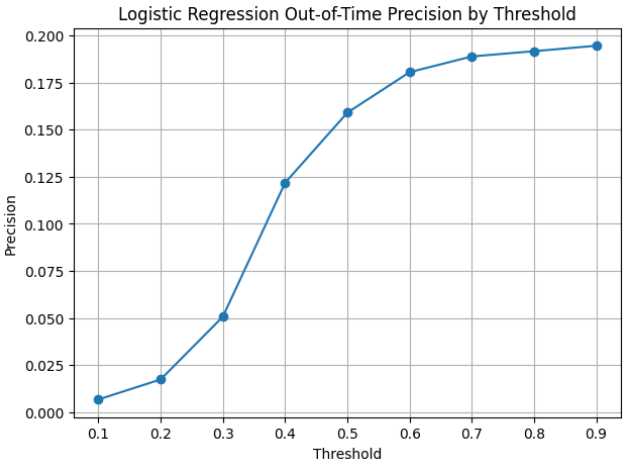

Fig. 4: Precision by threshold for the Logistic Regression model.

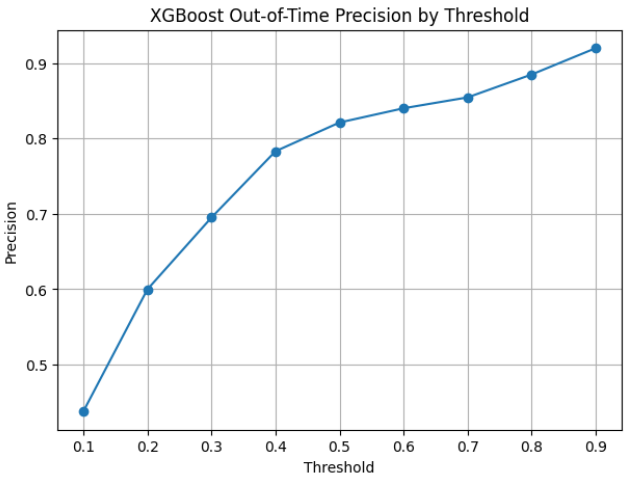

Fig. 5: Precision by threshold for the XGBoost model.

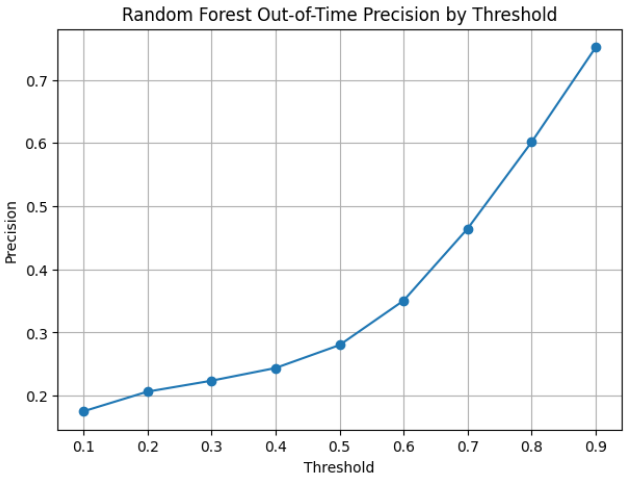

Fig. 6: Precision by threshold for the Random Forest model.

Figures 7, 8, and 9 show how recall changes across different classification thresholds for the three different models.

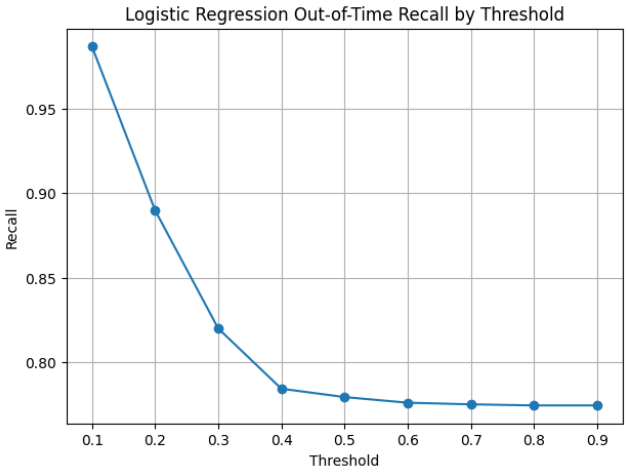

Fig. 7: Recall by threshold for the Logistic Regression model.

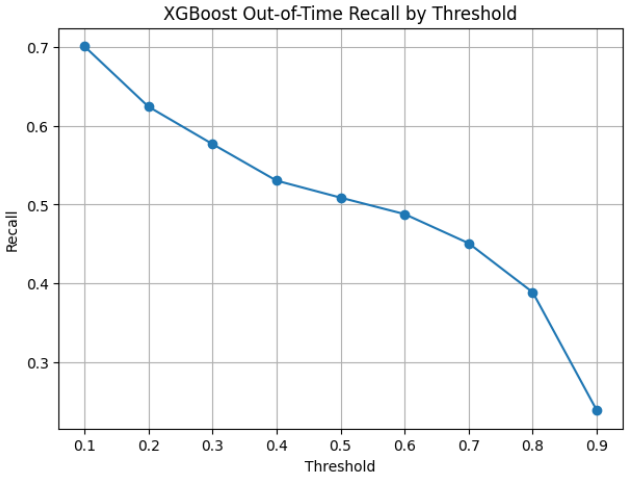

Fig. 8: Recall by threshold for the XGBoost model.

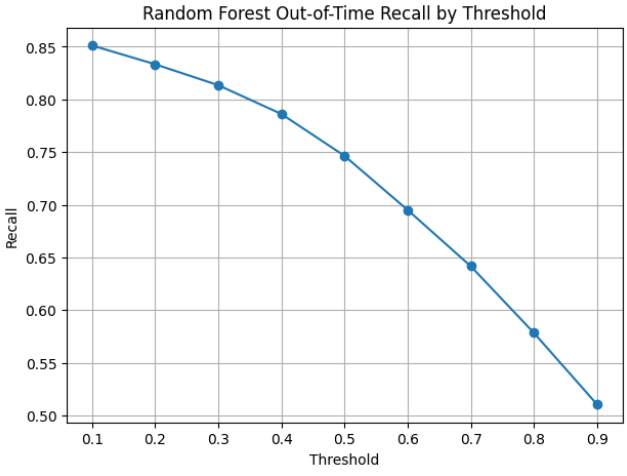

Fig. 9: Recall by threshold for the Random Forest model.

B. *Features Used in the Final Models*

- *Logistic Regression:* The final feature set was obtained by retaining only features that were consistently selected across all five LASSO subsamples, and then removing

one with high multicollinear VIF scores and one engineered feature that had the same explanatory power as its equivalent with another time window. 8 features were used for the final model:
  - 4 of which engineered features
  - 4 of which raw variables

  It can be noted that the features used by the Logistic Regression model include a smaller number of binary variables.
- *XGBoost:* After the feature selection process including SHAP values, 10 features were used:
  - 7 of which engineered features
  - 3 of which raw variables

  It can be noted that the features used by the XGBoost model include a majority of engineered variables and no binary variables.
- *Random Forest:* 10 features were used:
  - 6 of which engineered features
  - 4 of which raw variables

It can be noted that only one of these is a binary feature.

TABLE IV: Final Model Performance Comparison

Metric	$\text{XGBoost}_{\text{top10}}$	RF_{top10}	Log Reg
Precision@50	0.93	0.90	0.59
Precision@100	0.91	0.87	0.57
AUC (Out-of-Time)	0.95	0.94	0.92
Log Loss	0.01	0.04	0.26

VII. Discussion

A. Interpretation of the Results

The results show that it is in fact possible to get an overview of customer churn prediction using only financial data from the securities accounts segment. All models have an AUC-ROC score over 0.9 which means that they are highly useful for prediction. It is also noteworthy that the performance of the Logistic Regression model is relatively similar to that of the other models since XGBoost and Random Forest are both tree-based while Logistic Regression is linear. The relatively strong performance of the Logistic Regression model suggests that the selected financial features contain predictive patterns that are, to a meaningful extent, well approximated by a linear model. This is consistent with the argument by Levy et al. [19] that Logistic Regression should not be dismissed in favor of more complex methods when the underlying relationships are amenable to linear modeling. The recall for Logistic Regression, however, decreased significantly compared to the other models as the threshold increased. This is likely because logistic regression, as a linear model, mainly produces probabilities in line with the actual probabilities in the dataset, as described by van der Goorbergh et al. [20]. These are low as a result of the churned customers making up a small minority of the total number. Tree-based models on the other hand, and especially boosted ones, yield more extreme probabilities, according to Niculescu-Mizil and Caruana [21]. This suits the churn modeling perspective better due to the large majority of customers having an almost non-existent risk of churn while a very small number, actual potential churners, have a high probability of churning.

One thing worth noting is the properties of the "*Random Forest Out-of-Time Precision by Threshold*" figure compared to the equivalents by the other models. While those belonging to the Logistic Regression and XGBoost models have a more logarithmic shape, the Random Forest Precision curve looks more like an exponential curve. This could be as a result of the *Balanced Random Classifier* undersampling the majority class yielding a more aggressive flagging for the positive class.

Another interesting outcome of the feature selection step is the differences in features selected for each model. The number of binary variables selected from the LASSO selection for Logistic Regression was higher than those for the tree-based counterparts.

The VIF feature selection could also have eliminated continuous variables to a greater extent than binary variables, as continuous variables are generally more prone to multicollinearity. Binary variables, by their nature, tend to have lower correlations with one another since they can only take two values, making them less likely to be removed by a VIF-based selection process. This could partially explain the higher number of binary variables retained for the Logistic Regression model compared to the tree-based counterpart.

The reason for XGBoost performing the best overall out of the three models is likely, as mentioned, a result of tree-based models' ability to yield extreme probabilities in a way that is suitable for imbalanced datasets. A possible reason for XGBoost performing better than the other tree-based model, Random Forest, is likely the use of a gradient in the former when building the trees, giving XGBoost the advantage of learning sequentially. XGBoost could thus be seen as taking advantages both from Logistic Regression (gradient optimization for sequential learning) and Random Forest (tree properties to yield more suitable probabilities). The marginal difference in ROC-AUC scores however implies that both models have high performance.

The Log Loss values should be interpreted with caution as the value for XGBoost is extremely low which indicates overfitting as a result of class imbalance. This is also the case for Random Forest, although not as severely. The cause of this is likely the same as described referring to Niculescu-Mizil and Caruana [21] - that the tree-based models yield very high probabilities for actual churners, and practically non-existent ones for non-churners. Since non-churners make up most of the observations, this leads to a very low loss overall, similar to the accuracy paradox discussed in Section VII-B2.

B. Comments on Methodological Decisions made

1) Churn Definition: As with every study on customer churn, the choice of churn definition has an important impact on the end result. In this study, a trade-off between maximizing the number of observations used while still training the model on actual relevant customers had to be made. A large number of customers who were in the dataset but never active had to be removed in order for the definition to be implemented. This

can be defended using the reasoning that churn modeling is mainly relevant for customers who actually contribute to the business economically.

Many other studies on customer churn have been made on subscription services, which is fundamentally different since the customers who have canceled or not renewed their subscription have churned while those who still pay for the service are still active. The complexity of churn definition when the data consists of both activity-based variables, binary variables as well as level variables made it necessary to design the churn definition in a way that defined an actual relevant customer, which in itself is not particularly straightforward.

Using *Economic Churn* as a combination of inactivity and low portfolio market value can be viewed as a relatively conservative churn definition, since this prevents both active traders with low portfolio values (below SEK 500) and passive savers (active less frequently than every 6 months) with a more significant portfolio market value from being defined as churned. This is likely to be beneficial for the bank since there is a considerable number of customers who are only transactionally active annually.

2) Evaluation Measures: When developing a machine learning model, the decision on evaluation measures used is crucial for the usefulness of the final model itself. The choice has to be made with the data, class distribution and also the practical area of use in consideration. Choosing the wrong evaluation metric can have significant consequences for the reliability of the analysis itself. An example of this is using accuracy as an evaluation metric for imbalanced datasets such as the one used in this study. Since a very small share of the total number of observations consists of the positive class, a model classifying all observations as negative will still yield a high accuracy. This will result in a model that performs well on paper while still essentially being useless for what it was constructed for.

When using customer churn models as a tool for analysis rather than executive decision making, there is more room for trade-off between the emphasis made on precision and recall respectively, as a false negative will not lead to the same consequences as it might in e.g. screening for diseases. A false positive on the other hand will not have prominent effects on the business either, as no customers deemed as churned are filtered away in any way.

The use of Precision, Recall, ROC-AUC, and Log Loss for the final model is explained by these measures all providing relevant information on model performance while still giving an honest image of the model's actual practical use.

3) Not Using Class Imbalance Adjustment for XGBoost: As mentioned in the methodology, hyperparameters for imbalanced data sets resulted in significantly improved model performance for the Logistic Regression and Random Forest algorithms while the opposite was the case for XGBoost. The reason behind this is likely the second-order approximation in the regularized objective function as described by Chen and Guestrin [11] in the "*Theory on Machine Learning Methods used*" section where g_i and h_i are the first and second order gradients of the loss function. The first gradient describes the direction in which the error should be corrected, while the second gradient describes how confident the algorithm is in its prediction, causing the correction to be smaller for confident correct predictions and larger otherwise. The regularization term $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ penalizes tree complexity, which causes the model to be less prone to overfitting and thereby less fitted to the majority class. These properties are in this thesis shown to be an effective way of self-calibrating to deal with class imbalance.

C. Target Variable Construction and its Implications for Performance Metrics

The AUC score for all three models being over 0.9 and in particular the AUC score for XGBoost being 0.95 as well as the Log Loss scores being ¡ 0.05 for the tree-based models should be interpreted with caution. These numbers are likely a consequence of the construction of the target variable. Because the *Deactivated* churn type is defined by the disappearance of a customer ID at time *t+1*, part of the positive class could be trivially predictable from features that mechanically trend downward in the months leading up to deactivation rather than genuine behavioral signals. Since these metrics are computed across the full dataset, the trivially predictable *Deactivated* cases may disproportionately inflate the aggregate scores, potentially masking weaker performance on the more business-relevant *Economic* churn (described in Section V-D) cases. Further work should consider reporting model performance separately for the two churn types, particularly since the drivers of *Economic* churn are likely to carry greater business relevance.

D. Limitations

One limitation of this thesis was the product-level data available. In order to attain a cross-product analysis with a perspective on bank-loyalty for banks offering many different products (such as everyday banking, savings accounts, mortgages, securities accounts etc.), customer-level information about the extent to which an individual is active in the different products would have been necessary. Since empirical observations combined with intuition suggest that there might be a case for this type of variable within the customer churn field, an important greater perspective regarding customer loyalty that could be useful in marketing strategy could not be formed.

The focus solely on securities accounts might however provide a more detailed picture of these variables as churn factors, which is interesting from a pure trading-related perspective.

E. Possible Use of Models Developed

The models developed in this thesis are shown to perform relatively well in predicting customers that might churn, making the implementation of a "Top k" direct marketing strategy towards customers that are at high risk of churning a possible way to improve customer retention.

Despite the XGBoost model performing better than the other two in the evaluation metrics chosen, Logistic Regression could still be more useful as its interpretability could be seen as a clear advantage, especially since it is relatively close in terms of performance.

F. Bank Customer Loyalty

The enhancement of churn prediction for customer retention combined with the importance of customer retention for business success mentioned in the literature review makes churn prediction an important piece in marketing strategy. When also taking the "one stop shopping experience" for cross-buying emphasis by Liu and Wu [6] into account, successfully predicting customer churn for one segment might lead to a spillover effect resulting in improved customer loyalty on a complete bank level. Furthermore, customer churn prediction has strong connections to Customer Lifetime Value due to this relationship, giving the churn prediction additional relevance beyond the segment itself.

The importance of customer retention in itself mentioned by Barr and Pettersson [3] further motivates analyses of this kind, since prior research indicates that acquiring new customers is generally more costly than retaining existing ones.

The contradictory findings about retail bank customer loyalty mentioned in the literature review imply that the securities segment could be especially relevant but also vulnerable for more established multi-segment banks in the future.

G. Ethical considerations

The practice of modeling customer behavior in general can be viewed as ethically problematic, as it inherently requires the collection of large amounts of customer data. This raises questions about to what extent such extensive data collection is proportionate to its intended purpose, particularly when the models are used to draw conclusions about an individual's future behavior based on patterns in their past activity. This concern is further amplified when data linked to the individual at this level of granularity is used in machine learning models with lower explainability, such as the Random Forest and XGBoost models used in this thesis, where the specific reasoning behind a given prediction is considerably more difficult to trace back to the customer data than in more interpretable models such as Logistic Regression. Models of this kind are not exclusive to the finance sector, but are increasingly used in customer analysis across many sectors, raising questions about the moral boundaries regarding data usage for black-box algorithmic analyses.

VIII. Conclusion

The conclusion with regards to the research questions is that it is possible to predict customer churn using transaction data and that engineered features improve model performance when doing so. XGBoost was the best performing model of the three used in the analysis, especially when measured using Precision@k, likely due to its properties as a tree-based model. However, all three models are deemed sufficient for churn prediction according to the evaluation metrics used in this thesis. Logistic Regression has the advantage of being more interpretable than tree-based models, possibly making it more useful in practice.

Churn prediction constitutes an important component of customer retention strategy in retail banking, and is particularly relevant in the securities segment given the potential for spillover effects on customer loyalty across the bank's entire product offering. A central challenge for churn prediction lies in defining churn for non-contractual services, where the boundary between inactivity and attrition is inherently ambiguous. Any organization seeking to predict churn must therefore place considerable emphasis on tailoring the churn definition to its specific business context.

IX. Use of AI

AI (Claude, Anthropic and ChatGPT, Open AI) was used in this thesis in the following ways: (1) to identify and debug errors in the code; (2) for proofreading, identifying formatting inconsistencies, and refining formulations in the written text; and (3) as a sounding board to deepen understanding of theoretical concepts included in the thesis. All research design, problem formulation, methodological decisions, modeling, analysis, and interpretation of results were conducted by the authors. All AI-generated output was critically reviewed, and all changes to the text and code were made by the authors.

X. Individual contributions

Both authors actively contributed to the conception and development of this thesis. M. Occasus took primary responsibility for data processing, feature engineering, and feature selection, while E. Norén took primary responsibility for the written presentation, literature review, and analysis. All work was carried out collaboratively, with both authors involved in the design, interpretation, and refinement of each component.

References

[1] M. Haenlein, A. M. Kaplan, and A. J. Beeser, "A model to determine customer lifetime value in a retail banking context," *European Management Journal*, vol. 25, no. 3, 2007.

[2] Y. Ahn, D. Kim, and D.-J. Lee, "Customer attrition analysis in the securities industry: A large-scale field study in Korea," *International Journal of Bank Marketing*, vol. 38, no. 3, 2020.

[3] K. Barr and H. Pettersson, "Predicting and explaining customer churn for an audio/ebook subscription service using statistical analysis and machine learning," Master's thesis, 2019.

[4] J. B. G. Brito, G. B. Bucco, R. Heldt, J. a. L. Becker, C. S. Silveira, F. B. Luce, and M. J. Anzanello, "A framework to improve churn prediction performance in retail banking," *Financial Innovation*, vol. 10, no. 1, p. 17, 2024.

[5] T. Njaya and B. T. Sithole, "A comparative analysis of the determinants of customer churn behaviour in various industries in Asia, Latin America and Sub-Saharan Africa," 2021.

[6] T.-C. Liu and L.-W. Wu, "Customer retention and cross-buying in the banking industry: An integration of service attributes, satisfaction and trust," *Journal of Financial Services Marketing*, vol. 12, no. 2, pp. 132–145, 2007.

[7] Global Finance Magazine. (2026) Fintech surge drives Nordic bank M&A. [Online]. Available: https://gfmag.com/capital-raising-corporate-finance/fintech-surge-drives-nordic-bank-ma/

[8] J. Karlberg and M. Axén, "Binary classification for predicting customer churn," Master's thesis, 2020.

[9] L.-S. Yang and C.-C. Chiu, "Knowledge discovery on customer churn prediction," *Expert Systems with Applications*, vol. 31, pp. 515–524, 2006.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.

[11] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[12] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.
[13] A. Cerqua, M. Letta, and G. Pinto, "On the (mis)use of machine learning with panel data," 2025.
[14] J. He and M. X. Cheng, "Weighting methods for rare event identification from imbalanced datasets," *Frontiers in Big Data*, vol. 4, p. 715320, 2021.
[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[16] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," Univ. California, Berkeley, Tech. Rep. 666, 2004.
[17] Scikit-learn developers. (2024) `sklearn.utils.class_weight.compute_class_weight`. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html
[18] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.
[19] J. N. Levy, M. O'Brien, L. Tiwari, A. Bhatt, and Z. S. Grinspan, "Don't dismiss logistic regression: The case for sensible extraction of interactions in the era of machine learning," *BMC Medical Research Methodology*, vol. 20, no. 1, p. 171, 2020.
[20] R. W. M. van den Goorbergh, M. van Smeden, T. P. A. Debray, and J. A. H. de Groot, "The harm of class imbalance corrections for risk prediction models: Illustration and simulation using logistic regression," *Journal of Clinical Epidemiology*, vol. 150, pp. 156–164, 2022.
[21] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005, pp. 625–632.

TRITA-EECS-EX-2026:377
Stockholm, Sweden 2026

www.kth.se