

# PREDICTING CUSTOMER CHURN AND CUSTOMER LIFETIME VALUE (CLV) USING MACHINE LEARNING

MAGNUS GERDE

Master's thesis  
2023:E8



LUND UNIVERSITY

Faculty of Engineering  
Centre for Mathematical Sciences  
Mathematical Statistics

Master's Theses in Mathematical Sciences 2023:E8  
ISSN 1404-6342  
LUTFMS-3466-2023  
Mathematical Statistics  
Centre for Mathematical Sciences  
Lund University  
Box 118, SE-221 00 Lund, Sweden  
<http://www.maths.lu.se/>

---

# **Predicting Customer Churn and Customer Lifetime Value (CLV) using Machine Learning**

---

Magnus Gerde  
Guided By - Elias Ghattas, Ecster AB  
and  
Prof. Andreas Jakobsson

Master's Thesis is carried out at Ecster AB.

## Abstract

In an evermore competitive environment for companies and business, predictive customer behaviour models can give companies a competitive edge over its competitors. Two such important predictive behaviour models are customer churn models and customer lifetime value (CLV) models. As it is more expensive for companies to acquire new customers rather than retaining existing ones, it is important for business to keep their existing customer base. Customer churn models can assist in retaining existing customers as they can identify patterns in customer engagements and behaviour which increase the risk of churning. These high risk customers can then proactively be targeted with personalized retention strategies. CLV models can assist companies with predicting revenues and identify areas where the company can improve to meet revenue goals.

In this thesis, three different popular machine learning algorithms were used to predict customer churn: logistic regression, random forest(RF) and support vector classifier (SVC). Moreover, two different regression algorithms were used to predict CLV: linear regression and support vector regression(SVR). The results showed that the SVC model and the logistic regression model had similar results, with the SVC model having slightly better performance metrics. Moreover, as the feature data was significantly correlated, the logistic regression model might not generalize as well to new data, compared to the SVC model. The random forest model was unstable across different evaluation sets, was to reluctant to classify customers as churned and had overall the worst performance of the three models. For the CLV models, the linear regression model was unable to accurately model the skewed distribution in spending patterns among the customers. Compared to a naive predictor, the linear regression model was only able to outperform in predicting which customer would stop generating revenue. For the customers who did not stop generating revenue, the linear regression model performed significantly worse. The SVR model could more accurately model CLV, outperforming the naive predictor across all ranges except the 1/8:th highest spending customers. The SVR model further significantly outperformed the linear regression model, except for predicting which customers would stop generate revenue, where the linear regression model was slightly better.

**Keywords**— Churn Prediction, CLV Prediction, SVM, Logistic Regression, Linear Regression, Random Forest Model

## **Acknowledgement**

I would like to thank my supervisor at Ecster, Elias Ghattas, for helping me with everything relevant to the thesis from setting up meetings with relevant stakeholders, providing important insights about data extraction and modelling decisions to providing constructive feedback on the report. I would further like to thank Ecster for giving me the opportunity to investigate this interesting research topic as well as giving access to relevant data. It has been very exciting to apply knowledge gained from my studies in a real world application.

I would also like to thank my supervisor at LTH, Andreas Jakobsson, for providing guidance through the thesis and providing constructive feedback.

Lastly, I would like to thank my coworkers at the analytics department at Ecster for being supportive, providing encouragement and inviting me to fun activities.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>CLV Data &amp; Churn Definition</b>	<b>3</b>
<b>3</b>	<b>Model Setup</b>	<b>5</b>
3.1	Missing Values . . . . .	5
3.2	Outlier Values and Feature Standardization . . . . .	6
3.3	Special Feature: Churn Score . . . . .	7
3.4	Special Feature: Churn Proximity Score . . . . .	8
<b>4</b>	<b>Churn Model</b>	<b>9</b>
4.1	Background . . . . .	9
4.1.1	Logistic Regression . . . . .	10
4.1.2	Random Forest Model . . . . .	11
4.1.3	Support Vector Classification . . . . .	12
4.2	Feature Selection . . . . .	15
4.2.1	One Variable Logistic Regression Analysis . . . . .	16
4.2.2	SelectFromModel . . . . .	20
4.2.3	Forward Stepwise Logistic Regression . . . . .	21
4.2.4	Conclusions . . . . .	22
4.3	Model A - Logistic Regression . . . . .	22
4.4	Model B - Random Forest Model . . . . .	25
4.5	Model C - Support Vector Classifier . . . . .	26
4.6	Discussion & Conclusions . . . . .	28
<b>5</b>	<b>CLV Model</b>	<b>30</b>
5.1	CLV Data . . . . .	30
5.2	Background . . . . .	33
5.2.1	Linear Regression . . . . .	33
5.2.2	Support Vector Regression . . . . .	34
5.2.3	Scikit Method - GridSearchCV . . . . .	35
5.3	Model A - Linear Regression . . . . .	35
5.4	Model B - Support Vector Regression . . . . .	39
5.5	Discussion & Conclusions . . . . .	42

# Chapter 1

## Introduction

Over the recent years, the amount of data businesses and cooperations have available have skyrocketed. The vast amounts of data offers companies opportunities to utilize statistical- and machine learning models to make data-driven and more efficient decisions, rather than basing on personal biases or intuition. In an online survey about big data use cases conducted by BARC, in which 559 business and IT decision-makers participated, numerous benefits were found [1]. The biggest benefits include better strategic decisions (69%), improved control of operational processes (54%), a better understanding of customers (52%) and cost reductions (47%). Furthermore, those organizations able to quantify their gains from analyzing big data reported an average 8% increase in revenues and a 10% reduction in costs.

One application of big data analytics is predictive customer behavioral models such as churn and customer lifetime value (CLV) models. CLV describes the estimated net revenue a customer generates across the whole customer journey whereas churn is a business term describing when a customer stop doing business with their company. The advantages of accurate churn- and clv models are several:

- It can help companies identify customers who are at high risk of churning by analysing their past behaviours and engagements with the company. These high risk customers could then be proactively targeted with personalized retention strategies such as marketing.
- It assists companies in understanding their customer segments and helps identify which engagement strategies are the most efficient.
- Companies can predict revenues and identify areas where the company can improve in order to reach revenue goals.

This thesis is carried out at Ecster AB to investigate the efficiency of churn- and clv models. Ecster AB offers payment solutions for companies and private customers in Sweden and Finland. This thesis will specifically focus on Ecsters card and account solution for private customers in Sweden. The solution offers both credit card and partial payment plans for goods and services. Moreover, the models will focus on the time period 2021Q1-2022Q2 where data is taken on a quarterly basis.

This thesis will compare the applicability of classification algorithms logistic regression, random forest model and support vector classification for the churn model, predicting which customer will be in a churn respective non-churn state. Moreover, the thesis will compare regression algorithms linear regression and support vector regression for predicting customer lifetime value.



# Chapter 2

## CLV Data & Churn Definition

The customer lifetime value data is based on four different revenues: interest revenues, arrangement fees, administrative fees - account and administrative fees - payment plan. As the feature data points are captured on a quarterly basis, the revenues were summarized on a quarterly basis as well. For simplicity, the two different administrative fees, which are fees paid on a monthly basis, were bundled together and labeled monthly fees. Arrangement fees are paid when a customer initialise a payment plan and the value varies depending on the retailer and which payment plan the customer has. The value of monthly fees depends on the retailer and which type of payment plan the customer has. The interest revenues depends on the amount of borrowed money and the interest rate. The interest revenues is generated from both payment plans and products bought with unsecured credits. Although customer lifetime value is defined as the expected net revenue of a customer during their customer journey, the model will only model revenues.

In order to construct a model which aims to predict customer churn, a condition which defines when a customer has churned must be agreed upon. As the term loosely states a customer who has stopped doing business with the company, the term is a bit vague and the exact definition will vary from business case to business case. For this thesis, one approach would be to define customers who cancel their accounts as churned. However, the revenue data suggests there are customers with a non-cancelled account who are inactive and do not generate any revenue. To account for these inactive customers, the following definition is used for a churned customer:

Churn definition:

*Customers who cancel their account*

or

*has not made any payment nor transactions within 90 days.*

A payment means a customer has made a payment on an active payment plan whereas a transaction means a customer has made a transaction with their credit card. To justify the churn definition, table 2.1 is presented. The table shows revenues for customers who are active respective inactive with regards to payments and transactions. Active respective inactive transactions is defined as a customer who has- respective has not made a transaction within 90 days. Similarly, active respective inactive payments is defined as a customer who has- respective has not made a payment within 90 days.

<b>Customer Segment</b>	<b>Average Normalized Revenue - 2021Q1</b>	<b>Average Normalized Revenue - Full Period</b>
Active Transactions, Active Payments	1	1
Inactive Transactions, Inactive Payments	0.002	0.0399
Active Transactions	0.9515	0.9322
Active Payments	0.7138	0.6912
Inactive Transactions, Active Payments	0.5927	0.4306
Active Transactions, Inactive Payments	0.4948	0.4550

Table 2.1: Average revenues for different customer segments for 2021Q1 as well as the whole time period 2021Q1-2022Q2. To anonymize the data, the average revenues have been normalized by the largest spending segment, Active Transactions, Active Payments, for respective period. The customer segments are based on the activity of 2021Q1. Active transactions and inactive transactions are defined as customers who has respective has not made an transaction within 90 days. Similarly, active payments and inactive payments are defined as customers who has respective has not made a payment within 90 days.

Note that, by the churn definition, the customer segment inactive transactions and payments are defined as churned whereas the other customer segments are not classified as churned customers. Table 2.1 show that a customer who has been defined as churned based on 2021Q1 data only generates 0.002 the amount of the highest earning group for the first quarter of 2021. Furthermore, these customers who were classified as churned 2021Q1 generated much lower average revenues during the full time period 2021Q1-2022Q2, as it made only 0.0399 the amount of the highest earning group during the same period. This suggests that a customer who gets defined as churned will on average not be a future profitable customer.

Moreover, table 2.1 show that there exists customers who use both products(active transactions and payments) and those who only use one of the products(active payments, inactive transactions and inactive payments, active transactions) and that all of these customer segments are generating a significant amount of revenue on average. With the given churn definition, these revenue generating customer segments who use any of the products will not be defined as churned.

# Chapter 3


## Model Setup

The models were implemented to predict one quarter ahead. With one quarter foresight, there is sufficient time to act in order to mitigate customer churn or prevent predicted lower CLV. The total data were split up in three sets: model set, validation set and test set, consisting of 150000 random customers for both the model set and the validation set, and 100000 random customers for the test set. The customers were picked uniquely for each set, i.e. with no leakage. To prevent overfitting, the models were fitted on a certain quarter on the model set and then evaluated on another quarter on the validation respective test set. Specifically, the models were fitted on 2021Q2 on the model set. The fitted models were then evaluated using feature data from 2022Q1 to predict the outcome of 2022Q2 on the validation set. The final models were lastly evaluated using feature data from 2022Q1 to predict the outcome of 2022Q2 on the test set. To reduce the effects of unbalanced data sets, currently churned customers were omitted. Note that this is an approximation as a churn state is not terminal due to a customer can become non-churn by transitioning from being inactive to active. Moreover, to reduce the effects of unbalanced data sets for the clv model, customers who currently generates 0 revenue were omitted.

### 3.1 Missing Values

Missing data points is handled differently depending on data type and location of missing value. If a customer is missing data values across all parameters for a certain quarter, then this customer is assumed to not be registered for this particular quarter and is assumed to be churned. If data relating to churn classification, i.e. status relating to whether the account is cancelled, date of latest transaction and date of latest payment, is missing, then these data points are handled as shown in figure 3.1.

	Q1	Q2	Q3	Q4
Case A	T	NaN	NaN	F
Case B	F	NaN	NaN	T
Case C	T	NaN	NaN	T
Case D	F	NaN	NaN	F
Case E	NaN	NaN	T/F	T/F
Case F	T/F	T/F	NaN	NaN




	Q1	Q2	Q3	Q4
Case A	T	T	T	F
Case B	F	T	T	T
Case C	T	T	T	T
Case D	F	F	F	F
Case E	T	T	T/F	T/F
Case F	T/F	T/F	T	T

Figure 3.1: The figure illustrates how missing values in churn classification are handled in different cases. T, F, T/F represent the churn classification is true, false respectively can be either true or false. NaN represents a missing value.

For case A and B, the customer changed churn state somewhere between Q1 and Q4. As a pessimistic approach, the customer is assumed to churn as soon as possible, setting the missing values in these cases as churned. For case C, the customer has shown an inactive pattern surrounding the missing values and is therefore also assumed to churn for the missing values. Similarly for case D, the customer has shown an active pattern surrounding the missing value, and the missing value is therefore also assumed not to be churned. Missing values at the start or at the end of the timeline, i.e. case E and F, are handled pessimistically, assuming the customer has churned.

For missing values in feature parameters, these are also handled differently depending on location of missing value. A schematic figure illustrating how these missing values are handled is shown in figure 3.2.

	Q1	Q2	Q3	Q4
Case A	0.0	NaN	NaN	3.0
Case B	NaN	NaN	R	R
Case C	R	R	NaN	NaN



	Q1	Q2	Q3	Q4
Case A	0.0	1.0	2.0	3.0
Case B	0.0	0.0	R	R
Case C	R	R	0.0	0.0

Figure 3.2: The figure illustrates how missing values in feature parameters are handled in different cases. NaN represents a missing value and  $\mathbb{R}$  represent any real value.

If there are missing values in the middle of the timeline, then the missing values are linearly interpolated from its nearest non-missing values. If the values are missing at the start or end of the timeline, then the missing values are put to zero.

## 3.2 Outlier Values and Feature Standardization

To deal with outliers, the z-score is considered. The z-score measures how much a certain value deviates from the mean, where a z-score of 0 means the value is equal to the mean value,

whereas a z-score of 1 is one standard deviation away from the mean, cf. [2]. To reduce the effects of outliers, customers who has any feature value which is 5 or more standard deviations away from the mean, i.e. a z-score of 5, is removed. Similarly for the clv model, a customer which has a clv value with z-score of 5 or more is removed. Furthermore, since some machine learning algorithms make distance computations or minimization schemes, features with a larger spread in distribution will become more dominant than features with less spread. To account for this, the data is standardized. Standardization means each parameter is transformed to zero-mean with unit variance.

### 3.3 Special Feature: Churn Score

To quantify how active a certain customer has been during a certain period, the churn score feature is considered. The churn score is designed to evaluate a customer's activity over a longer time period while considering the most recent activity higher. The churn score is given by:

$$X_{CS}(t) = \frac{1}{X_{CS}^{max}} \sum_{i=0}^N \alpha^i X_{t-i}^{churn} \quad (3.1)$$

where  $\alpha \in [0, 1]$  and N is the number of points evaluated. Moreover,  $X_{t-i}^{churn}$  is given by:

$$X_{t-i}^{churn} = \begin{cases} 1, & \text{if churned in time instance } t - i \\ 0, & \text{if not churned in time instance } t - i \end{cases}$$

Lastly,  $X_{CS}^{max}$  is the maximum possible churn score, i.e;

$$X_{CS}^{max} = \sum_{i=0}^N \alpha^i$$

$\alpha=1$  means that a churn classification is considered equal, regardless of when the classification occurred whereas  $\alpha=0$  only takes the latest churn classification into consideration, yielding 0 for an active customer and 1 for a churned customer. The feature is further normalized by the maximum score, meaning the highest possible feature value is 1. The  $\alpha$  - and N parameters were set to 0.33 respective 2 for the models. To exemplify the feature, different output values for different inputs, with  $\alpha=0.33$  and N=2, are shown in the following table:

Example	Input	Output
A	$\vec{X} = [0 \ 0 \ 0]$	0.0
B	$\vec{X} = [1 \ 1 \ 1]$	1.0
C	$\vec{X} = [0 \ 1 \ 1]$	0.92
D	$\vec{X} = [1 \ 1 \ 0]$	0.31

Table 3.1: Example of churn score outputs, eq. 3.1, for different input values where  $\vec{X} = [X_{t-2}^{churn} \ X_{t-1}^{churn} \ X_t^{churn}]$ , N=2 and  $\alpha=0.33$ .

In table 3.1, example A and example B show customers who have been active the last three quarters respective churned the last three quarters and the respective outputs 0.0 and 1.0. The table

further exemplifies how recent churn classifications yields a greater churn score as example C and D have the same number of churn classifications but example C has more recent churn classifications, yielding a higher churn score.

### 3.4 Special Feature: Churn Proximity Score

Since a customer who has not made any payments nor transactions within 90 days is classified as churned, a feature which estimates how close a customer is to be classified as churned by inactivity can be defined:

$$X_{CPS}(\Delta_t^T, \Delta_t^P) = \left(\frac{1}{90} \min(90, \Delta_t^T)\right) \cdot \left(\frac{1}{90} \min(90, \Delta_t^P)\right) \quad (3.2)$$

where T is transaction, P is payment,  $\Delta_t^T$  and  $\Delta_t^P$  are the number of days since the latest transaction respective payment. Similarly to the churn score feature, the churn proximity score has the range of [0,1] where 1 represents a churned customer (90 days or more inactive with regards to both transactions and payments) and 0 has made either a payment or a transaction today. The feature is designed to generate a high value if a customer is close to being classified as churned by inactivity whereas a customer with a low value has recently been active. Similarly to the churn definition, a customer must be inactive with regards to both transactions and payments in order to yield a high churn proximity score. To justify the feature definition, consider the following example of inputs and outputs:

Example	Input	Output
A	$\vec{X} = [90 \ 0]$	0.0
B	$\vec{X} = [0 \ 90]$	0.0
C	$\vec{X} = [90 \ 90]$	1.0
D	$\vec{X} = [45 \ 45]$	0.25
E	$\vec{X} = [81 \ 90]$	0.9

Table 3.2: Example of churn proximity score outputs, eq. 3.2, for different input values where  $\vec{X} = [\Delta_t^T \ \Delta_t^P]$ .

Example A and B in table 3.2 represents customers who are currently only active with respect to transactions respective payments. As these customers are active, the resulting churn proximity score is 0. Example C is a churned customer, as it has been inactive with regards to both payments and transaction for 90 days and yields an output of 1. Example D represent a customer who has not made any transactions nor payments within 45 days, resulting in a higher churn proximity score. Example E is only 9 days away of being inactive with regards to payments to be classified as churned, yielding an even higher churn proximity score.

# Chapter 4

## Churn Model

Three different algorithms will be used to predict customer churn: logistic regression, random forest model, as well as support vector classification.

### 4.1 Background

To compare the performance of different algorithms, some measures used to compare different classification algorithms is presented in table 4.1, cf. [3]:

Measure	Explanation	Equation
True Negative(TN)	A negative outcome which was predicted as negative.	-
False Positive(FP)	A negative outcome which was predicted as positive.	-
False Negative(FN)	A positive outcome which was predicted as negative.	-
True Positive(TP)	A positive outcome which was predicted as positive.	-
Accuracy	Fraction of outcomes which was correctly predicted	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Fraction of true predictions which was correctly predicted.	$\frac{TP}{FP+TP}$
Sensitivity	Fraction of true samples which was correctly predicted.	$\frac{TP+FN}{TP+FN}$
Specificity	Fraction of negative samples which was correctly predicted	$\frac{TN}{TN+FP}$

Table 4.1: Some measures with corresponding explanations and equations used to compare the performance of different classification algorithms.

To investigate correlations between features, the Pearson's correlation can be used, cf. [4]:

$$Correlation(X, Y) = \frac{S_{xy}}{S_x S_y} \quad (4.1)$$

where X and Y are two separate feature vectors,  $S_x$  and  $S_y$  is the standard deviation of X and Y respectively and  $S_{xy}$  is the covariance of X and Y, which is given by:

$$S_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (4.2)$$

The Pearson correlation gives a measure of the linear relationship between feature X and Y which is in the range [-1,1]. If two features X and Y have a Pearson correlation of 1, then there is a perfect positive correlation between the features, whereas -1 means perfect negative correlation. If the Pearson correlation is 0, then there is no linear relationship between the variables.

### 4.1.1 Logistic Regression

Logistic regression is a common method in predictive analytics and is used to predict a binary outcome, for example the churn state of a customer. The idea of logistic regression is that each binary outcome is the result of a binomial distribution with an individual probability,  $p_i$ , which in turn is dependent on some covariates, cf. [5]. The logistic regression problem can be formulated as:

$$\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_n x_{in} \equiv \text{odds} \quad (4.3)$$

where  $p_i$  is the probability of a true outcome of the  $i$ :th element,  $x_{ij}$  is the  $j$ :th covariate of the  $i$ :th element and  $\beta_j$  are coefficients of the  $j$ :th covariate. Logistic regression further assumes that the covariates are independent. As each outcome is interpreted as an outcome of a binomial distribution, the observed data can be expressed as a log-likelihood function, cf. [6]:

$$\begin{aligned} \ln(L(\beta; \mathbf{Y})) &= \ln(\prod_i Pr(y_i; 1, p_i)) = \\ &= \ln(\prod_i p_i^{y_i} (1 - p_i)^{(1-y_i)}) = \sum_i y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \end{aligned} \quad (4.4)$$

where  $\mathbf{Y}$  is a vector with the binary outcomes and  $y_i$  is the binary outcome of the  $i$ :th element. Note that equation 4.4 is dependent on  $\beta$  as  $p_i$  is dependent on  $\beta$ , according to eq. 4.3. Rewriting equation 4.3 yields:

$$\ln\left(\frac{p_i}{1-p_i}\right) = \text{odds} \iff p_i = \frac{e^{\text{odds}}}{1 + e^{\text{odds}}} \quad (4.5)$$

Inserting eq. 4.5 in eq. 4.4 yields a likelihood problem which is dependent on the  $\beta$  - coefficients. These  $\beta$  - coefficients are then selected to maximize the likelihood of observing the observed data, cf. [6], i.e. solving:

$$\beta = \underset{\beta}{\operatorname{argmax}} \ln(L(\beta; \mathbf{Y})) \quad (4.6)$$

This is a non-linear optimization problem and is solved with some appropriate iteration scheme. The optimal  $\beta$  - coefficients are selected by solving eq. 4.6 on a training set. These are then used to compute probabilities of churn, using eq. 4.5, giving an output between 0 and 1. All customers who have a probability of a certain threshold or above is predicted to churn whereas customers who have a probability less than the given threshold is interpreted to not churn.

Moreover, there are methods to measure how well a model fits to the data. One such measure is to compare the values of the optimal log-likelihood function, eq. 4.4, where a model with a higher log-likelihood function value have a higher probability to explain the observed data and hence is a better model. Another approach is to use pseudo  $R^2$ , given by [7]:

$$R^2 = 1 - \frac{\ln(L(M_{full}))}{\ln(L(M_{intercept}))} \quad (4.7)$$



where  $M_{full}$  is the full model, containing all parameters, and  $M_{intercept}$  is a model only containing an intercept. The idea of the method is to compare how much of the data is explained by the full model, compared to only the intercept. If the full model is significantly much better than the intercept model, then;

$$\begin{aligned} |\ln(L(M_{full}))| \ll |\ln(L(M_{intercept}))| &\implies \\ \implies \frac{\ln(L(M_{full}))}{\ln(L(M_{intercept}))} \approx 0 &\implies R^2 \approx 1 \end{aligned}$$

On the other hand, if the full model is not significantly much better than the intercept model, then;

$$\begin{aligned} |\ln(L(M_{full}))| \approx |\ln(L(M_{intercept}))| &\implies \\ \implies \frac{\ln(L(M_{full}))}{\ln(L(M_{intercept}))} \approx 1 &\implies R^2 \approx 0 \end{aligned}$$

### 4.1.2 Random Forest Model

Random forest model is a common classification algorithm used in machine learning applications due to its speed, flexibility and robust approach to analyze high dimensional data, cf. [8]. The foundation of a random forest model are decision trees. The idea of a decision tree is to divide the data set into smaller subsets, with the goal of minimizing the mixture in each subset, cf. [9]. Each subset is then assigned a prediction, based on the training data. A schematic figure illustrating a decision tree with three features and with depth two is shown in figure 4.1.

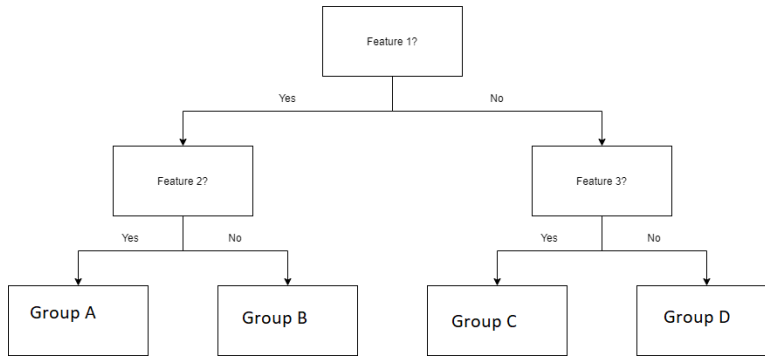


Figure 4.1: The figure illustrates a decision tree with three features and with a depth of two [10].

In the above example, the data set is partitioned in two different subgroups, based on an if-else statement with regard to feature 1. The yes branch is then evaluated with another if-else statement with regard to feature 2 whereas the no branch is similarly partitioned with an if-else statement with regard to feature 3. Each group - A,B,C and D is then assigned a prediction based on majority voting in the given group. The if-else statement in each data split is selected such that the decrease in Gini impurity is maximized, cf. [8]. The Gini impurity is given by:

$$\hat{\Gamma}(t) = \sum_{j=1}^J \hat{\phi}_j(t)(1 - \hat{\phi}_j(t)) \quad (4.8)$$

where  $\hat{\phi}_j(t)$  is the class frequency for class  $j$  in the node  $t$ . The decrease in Gini impurity with regards to a certain feature is computed by taking the difference of the Gini impurity before the split, i.e. the parent node, and the weighted sum of the Gini impurities of the two child nodes. The Gini impurity can further be used to assign an impurity importance of a certain feature  $X_i$ . The impurity importance is computed by summarizing the impurity decrease for all nodes which conducted a split w.r.t  $X_i$  and normalizing by number of trees. A drawback with the Gini impurity is that it tends to prefer continuous variables over binary variables, cf. [8].

A decision tree is applicable in customer segmentation applications, such as predicting churn, as it naturally divides all customers into different customer segments. An issue with decision trees, however, is that they tend to overfit to training data, cf. [11]. To overcome this issue, random forest model can be used. In a random forest model, many decision trees with identical structure are created in parallel where each decision tree is assigned a random subset of features and a random subset of the data, cf. [8]. Each decision tree then provides a prediction based on the features and training data it has been provided. The final prediction is then selected by majority voting, where the decision which were selected by the most decision trees is chosen as the final prediction. A schematic example of a random forest model is shown in figure 4.2.

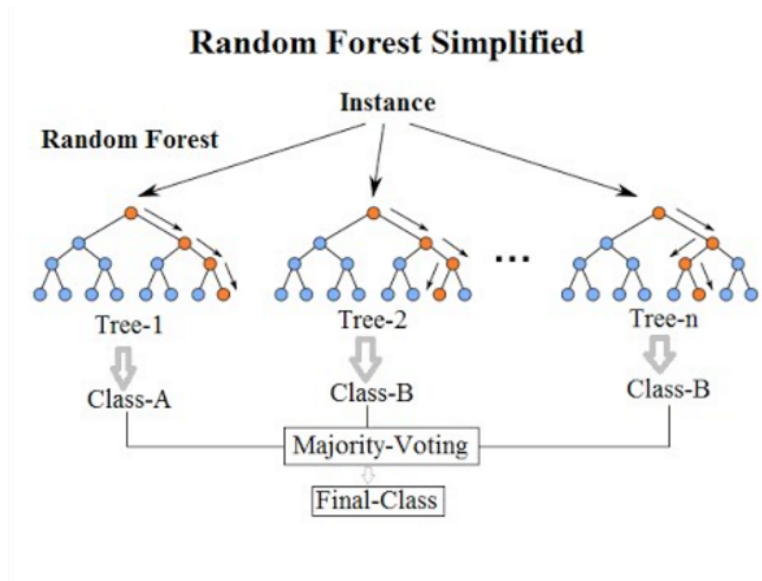


Figure 4.2: The figure illustrates a random forest model with decision trees with depth three [12].

An assumption of the random forest model is that the provided features have some predictive power, cf. [13]. Moreover, the decision trees needs to be uncorrelated, which in turn is affected by the provided features and hyperparameters.

### 4.1.3 Support Vector Classification

Support vector classification (SVC) is a supervised algorithm developed at the ATT Bell Laboratories by V. Vapnik et. al in 1995. It is widely used in machine learning applications such as

image analysis, biological applications and churn models, for its robustness and high accuracy, cf. [14]. Support vector classification tries to find natural clusters in features by separating the data in hyperplanes. To exemplify this, consider figure 4.3.

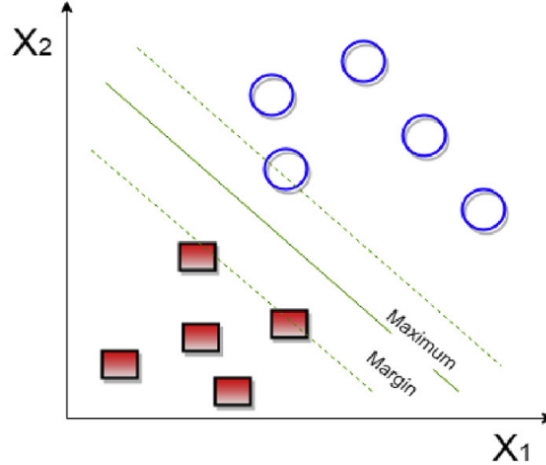


Figure 4.3: The figure illustrates the support vector classification algorithm [15]. The circle and square represents different classes to predict.

The aim of the support vector classification algorithm is to find the hyperplane which maximises the margin between the clusters, cf. [16]. When trying to predict new data point, the algorithm evaluates which side of the hyperplane the data points fall into, predicting a circle if it falls outside of the hyperplane or a square if it falls within the hyperplane. Mathematically, the data set

$$(x_1, y_1), (x_2, y_2) \dots (x_n, y_n), y_i = \{-1, 1\}$$

is said to be linearly separable if there exists a vector  $\mathbf{w}$  such that:

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geq 1, & \text{if } y_i = 1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1, & \text{if } y_i = -1 \end{cases} \quad (4.9)$$

where  $y_i$  is the class label, either -1 or 1,  $x_i$  is the feature value and  $b$  is the intercept of the separating hyperplane. The support vector classification algorithm is given by:

$$\underset{\mathbf{w}}{\text{minimize}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum \epsilon_i \quad (4.10)$$

subject to:

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \epsilon_i, \quad i=1,2,3\dots \\ \epsilon_i &\geq 0, \quad i=1,2,3\dots \end{aligned} \quad (4.11)$$

where  $C$  is a hyperparameter. To account for situations when the data cannot be neatly separated, a slack variable  $\epsilon_i$  is added to allow for certain outliers, see figure 4.4 for an example. It can be shown that the first term of the minimization problem is the same as maximizing the

separating hyperplane. The second term allows for outliers, however, these should be minimized as much as possible. The constraints is a reformulation of eq. 4.9, with the addition of slack variable  $\epsilon_i$  to allow for outliers. The constraints express that the hyperplane needs to be selected such that the label classes are correctly separated, with some exceptions. For large choices of the C-parameter, the classifier will allow for fewer outliers, prioritizing separating the classes even if it yields a lower separating margin. For lower C-parameters, on the other hand, it will prioritize a larger separating plane, even if it means that more labels are misclassified.

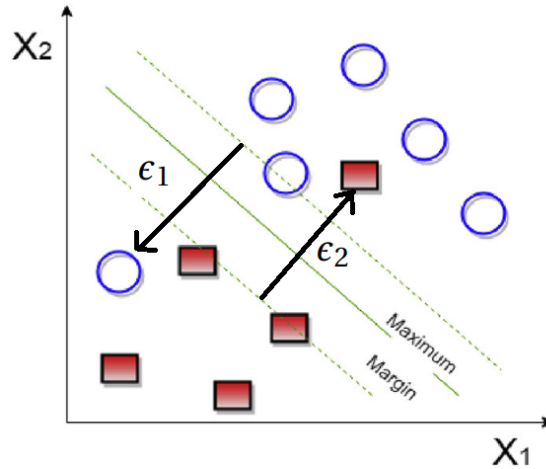


Figure 4.4: The figure illustrates the support vector classification algorithm with soft margin hyperplane [15]. The circle and square represents different classes to predict.

Moreover, a kernel trick can be used to improve the accuracy of the support vector classification algorithm, even when there are no natural choices of separating hyperplanes, cf. [14]. The kernel trick works by mapping the points to a higher dimensional plane, where an appropriate separating hyperplane exists. See figure 4.5 for an example of a kernel trick.

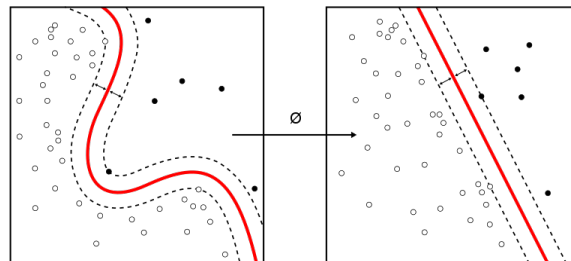


Figure 4.5: The figure illustrates a kernel trick [17]. In the left figure, no separating hyperplane exists. However, after applying a certain mapping  $\theta$ , a separating hyperplane exists.

An example of a kernel trick is the radial basis function (RBF), which is given by [18]:

$$\theta_{RBF} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} \quad (4.12)$$

where;

$$k(x_i, x_j) = e^{-\frac{d(x_i, x_j)^2}{2l^2}} \quad (4.13)$$

where  $l$  is the length scale of the kernel and  $d(\cdot, \cdot)$  is the Euclidian distance.

On the other hand, noisy data can lead to difficulties in finding a separating hyperplane and can lead to an underperforming algorithm. Moreover, the support vector classifier is a computationally expensive algorithm and therefore does not scale well with larger data sets, cf. [19]. It further does not perform well with unbalanced data sets.

## 4.2 Feature Selection

Given the large state space, with roughly 70-80 parameters, it is important to eliminate irrelevant variables to improve model accuracy and decrease execution time. In this thesis, numerous approaches are used: one variable logistic regression analysis, stepwise forward regression using logistic regression and the scikit-learn method `SelectFromModel()` [18] using two different classification algorithms RF and linear SVC. A selection of all variables are shown in table 4.2. The table shows variable label, explanation and whether the variable is continuous or binary.

Table 4.2: A selection of all variables. The table shows label, explanation and whether the variable is continuous or binary.

Label	Explanation	Variable type
$x_{HAB}$	Highest account balance	Continuous
$x_{UC}$	Used credit	Continuous
$x_{NUC}$	Notified used credit	Continuous
$x_{AP}$	Accumulated payments	Continuous
$x_{OAB}$	Outgoing account balance	Continuous
$x_{IAB}$	Ingoing account balance	Continuous
$x_{LAP}$	Lowest amount to pay (proportional to total credit debt)	Continuous
$x_{CP}$	Churn Proximity Score	Continuous
$x_{CS}$	Churn Score	Continuous
$x_{IN}$	Incremental number	Continuous
$x_{TR}$	Total Revenue	Continuous
$x_{LAC}$	Latest agreement change	Continuous
$x_{GC}$	Granted credit	Continuous
$x_{AR}$	Accumulated rent paid	Continuous
$x_{PPA}$	Pre-paid amount	Continuous
$\theta_{PP}$	Have a payment plan	Binary
$\theta_{PIA}$	Pays one or more invoice in advance	Binary
$\theta_{DF}$	Use digital bill instead of analog	Binary
$\theta_{RP}$	Have rest (delayed) payment(s)	Binary
$\theta_{PFM}$	Use payment free month - an offer to skip paying an invoice	Binary

### 4.2.1 One Variable Logistic Regression Analysis

Logistic regression offers different metrics to evaluate how well a model fits a binary prediction problem. Two such metrics are pseudo R-squared and log-likelihood. These can be used to get a priority list of the most relevant predictors. For the given data, a subset of all variables could be fitted as a simple linear model, eq. 4.14, reasonably well.

$$f(x) = \beta_0 + \theta(-x)\beta_1 + x\beta_2 \quad (4.14)$$

where;

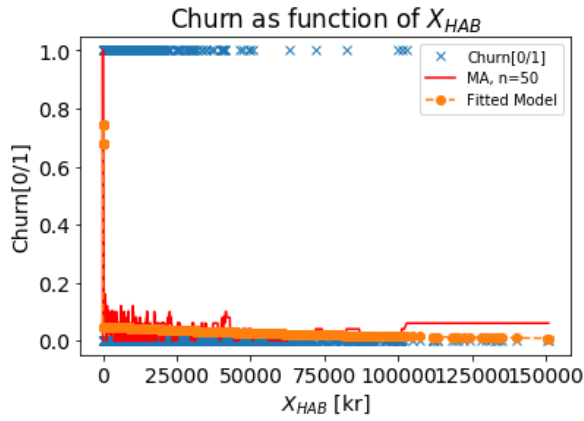
$$\theta(-x) = \begin{cases} 0, & \text{if } x \geq 0 \\ 1, & \text{if } x < 0 \end{cases}$$

The model presented above, equation 4.14, was iteratively fitted for each variable. Table 4.3 show the top 10 variables, sorted by pseudo r-squared:

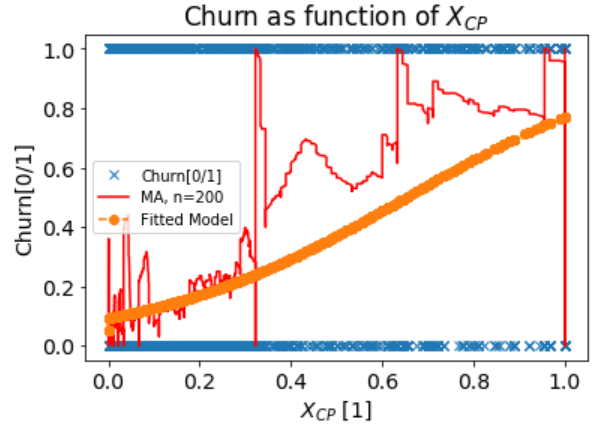
Table 4.3: Priority list of the most important variables, fitted by equation 4.14.

Label	Pseudo R-squared	Log-likelihood	$\beta_0$	$\beta_1$	$\beta_2$
$x_{HAB}$	0.4042	-7741.8	$-2.98 \pm 0.33$	$0.00 \pm 0.35$	$4.05 \pm 0.35$
$x_{HSFP}$	0.3476	-8476.7	$-2.22 \pm 0.27$	$0.00 \pm 0.29$	$3.15 \pm 0.3$
$x_{CP}$	0.3428	-8540.2	$-2.28 \pm 0.073$	$3.4943 \pm 0.095$	$-0.6201 \pm 0.211$
$x_{UC}$	0.3189	-8850.6	$-0.79 \pm 0.16$	$-0.00 \pm 0.23$	$2.05 \pm 0.19$
$x_{NUC}$	0.2841	-9302.4	$-0.77 \pm 0.15$	$-0.00 \pm 0.21$	$1.76 \pm 0.19$
$x_{AP}$	0.276	-9408.2	$-2.88 \pm 0.21$	$0.00 \pm 0.23$	$3.29 \pm 0.25$
$x_{OAB}$	0.2748	-9423.8	$-0.79 \pm 0.15$	$-0.0002 \pm 0.21$	$1.7087 \pm 0.19$
$x_{CS}$	0.2483	-9767.4	$-1.5895 \pm 0.107$	$2.7362 \pm 0.129$	$-0.1663 \pm 0.122$
$x_{IAB}$	0.2368	-9917.5	$-0.85 \pm 0.15$	$-0.0001 \pm 0.201$	$1.5431 \pm 0.186$
$x_{LAP}$	0.1533	-11002	$-2.59 \pm 0.28$	$0.0002 \pm 0.289$	$2.416 \pm 0.305$

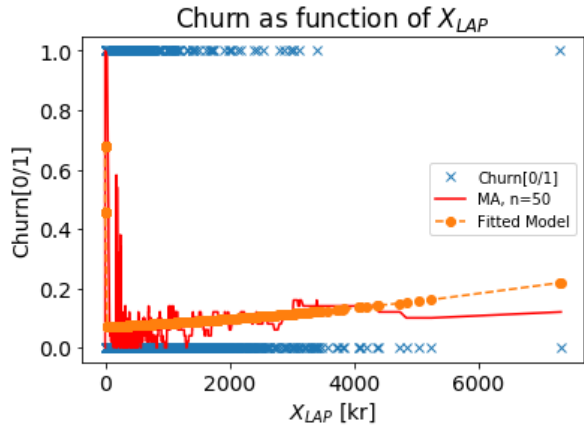
The proposed solution have the obvious problem that a variable can explain the data well with another choice of model, linear or non-linear. In fact, table 4.3 show that many models have a  $\beta_1$  variable which is not significantly different from zero and could therefore be excluded. Moreover, as multiple tests are conducted simultaneously, the risk of erroneous inferences increases. However, as the models are standardized and the models are one-dimensional, it makes it possible to visualize each variable separately. This gives an understanding of patterns in the data. Some of the most important variables, listed in table 4.3, are shown in figure 4.6.



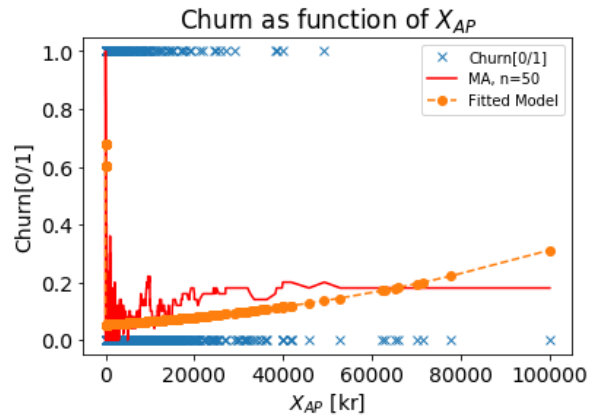
(a) Highest account balance.



(b) Churn proximity score.



(c) Lowest amount to pay.



(d) Accumulated payments.

Figure 4.6: Plots visualizing some of the most important variables in table 4.3. The plots show churn, where churned customers are represented as 1 and non-churned customers are represented as 0, as a function of each variable. To visualize the trend, a moving average with  $n=50$  and the fitted model, given by eq. 4.14, are shown.

In this somewhat tedious process, numerous important discoveries were made: Firstly, there are a large subset of variables with a similar distribution, as shown in figures 4.6a, 4.6c and 4.6d. Secondly, figure 4.6b show that customers with churn proximity closer to 1 tend to churn more, as the moving average is closer to 1. Since the feature is designed to tend towards 1 as a customer is almost classified as churned by inactivity, the trend is reasonable. Thirdly, figure 4.6a show a discontinuous jump in churn activity for customers who has a positive highest account balance compared to customers who has exactly zero as highest account balance. Perhaps this feature could be modeled as a binary model with positive highest account balance represented as 1 and 0 highest account balance represented as 0. Building further on this discovery of binary customer segments, a binary classification model was similarly fitted for each variable in the data. The binary classification model is shown in equation 4.15.

$$f(x) = \beta_0 + \theta(x)\beta_1 \quad (4.15)$$

where;

$$\theta(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$$

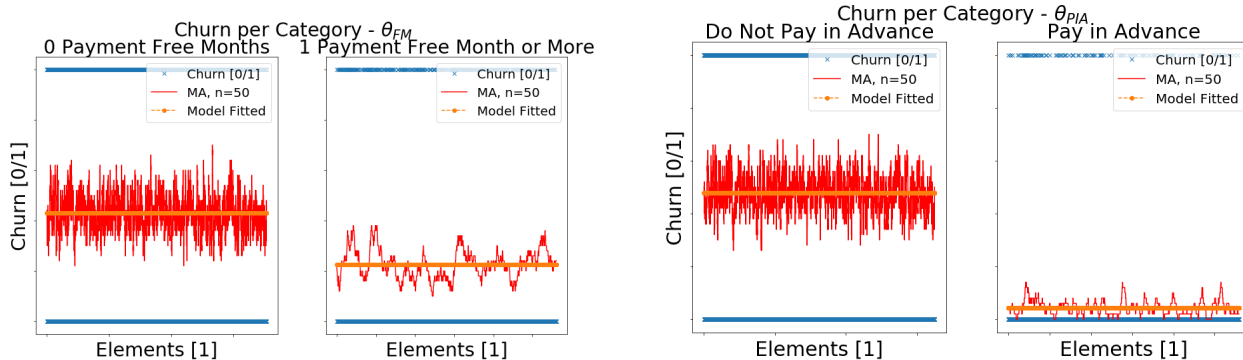
A priority list of the top 5 most important binary variables, sorted by log-likelihood, are shown in table 4.4.

Table 4.4: Priority list of the most important variables, fitted by equation 4.15.

Label	Pseudo R-squared	Log-likelihood	$\beta_0$	$\beta_1$
$\theta_{HAB}$	0.4017	-7746	$0.9278 \pm 0.043$	$-40832 \pm 0.115$
$\theta_{PP}$	0.135	-11245	$0.104 \pm 0.033$	$-2.598 \pm 0.112$
$\theta_{PIA}$	0.092	-11792	$-0.083 \pm 0.031$	$-3.023 \pm 0.184$
$\theta_{DF}$	0.018	-12761	$-0.229 \pm 0.031$	$-1.031 \pm 0.100$
$\theta_{RP}$	0.000	-12988	$-0.349 \pm 0.029$	$0.648 \pm 0.383$
$\theta_{PFM}$	0.009	-12871	$-0.282 \pm 0.030$	$-0.947 \pm 0.127$

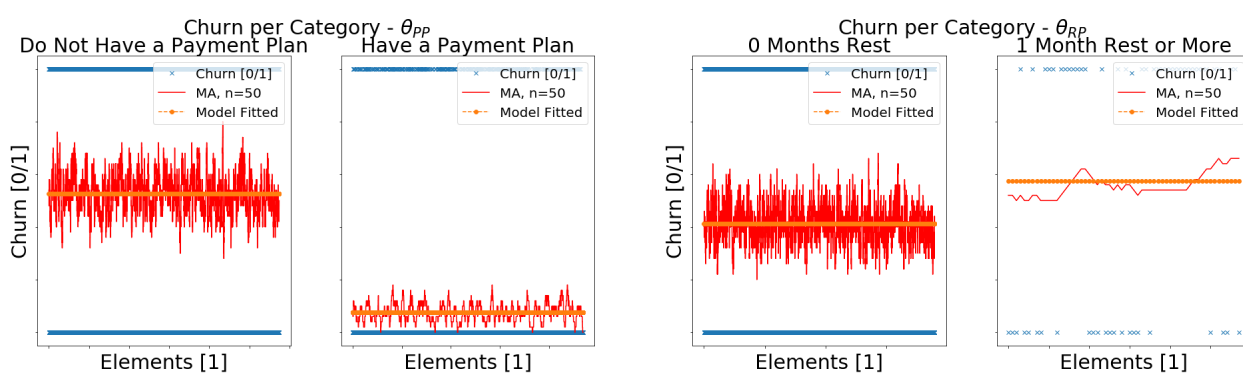
Notably by comparing R-squared values, the continuous version and the binary version of the highest account balance feature,  $x_{HAB}$  in table 4.3 and  $\theta_{HAB}$  in table 4.4, have similar R-squared values with 0.4042 and 0.4017 for  $x_{HAB}$  respective  $\theta_{HAB}$ . Some of the most important binary variables are shown in figure 4.7. The figures show the churn distribution of customers when the variable is equal to zero, left figures, as well as the churn distribution of customers when the variables are positive, right figures.





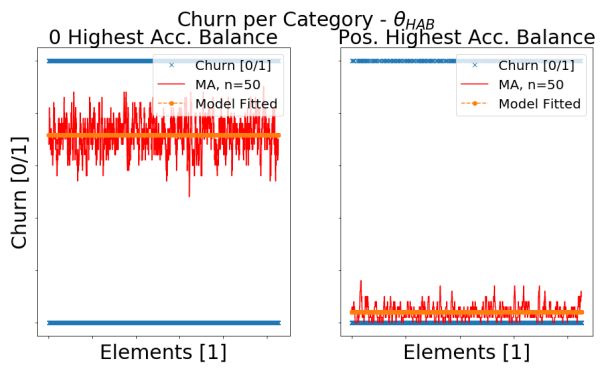
(a) Payment free month(s).

(b) Pay in advance.



(c) Have a payment plan.

(d) Have rest (delayed) payment(s).



(e) Highest account balance.

Figure 4.7: Plots visualizing some of the most important variables in table 4.4. The plots show churn, where churned customers are represented as 1 and non-churned customers are represented as 0. The left and right figures illustrates the churn distribution when the variable is equal to zero respective positive. To visualize the trend, a moving average with  $n=50$  and the fitted model, given by eq. 4.15, are shown.

Figure 4.7 show many interesting customer segments. Firstly, it visualizes what have previously been noted, namely that customers who have a positive highest account balance is significantly more active than customers who have exactly zero highest account balance. It further shows that customers who use available customer offerings such as payment free month, pay in advance and has a payment plan, tends to be more active. Lastly, the figure suggests that cus-

tomers who have paid one or more invoice too late tend to churn more.

Investigating the variables in table 4.3, it is evident that many variables are explaining virtually the same phenomena. For example, the variables  $x_{HAB}$ ,  $x_{IAB}$  and  $x_{OAB}$  which are highest-, ingoing and outgoing account balance, are all explaining very similar things and are therefore likely to be heavily correlated. When selecting features, it is important to take into consideration how the features correlates to each other. Many machine algorithms performs worse with correlated variables, especially logistic regression which requires uncorrelated feature variables. Table 4.5 show the correlation, using Pearson correlation, of the continuous variables found in table 4.3. Similarly, table 4.6 show the correlation, using Pearson correlation, of the binary variables found in table 4.4.

Table 4.5: Correlation, using Pearson's coefficients, of the continuous variables found in table 4.3.

	$x_{HAB}$	$x_{CP}$	$x_{UC}$	$x_{NUC}$	$x_{AP}$	$x_{OAB}$	$x_{CS}$	$x_{IAB}$	$x_{LAP}$
$x_{HAB}$	1.00	-0.53	0.87	0.83	0.26	0.81	-0.49	0.78	0.53
$x_{CP}$	-0.53	1.00	-0.46	-0.46	-0.26	-0.46	0.89	-0.45	-0.32
$x_{UC}$	0.87	-0.46	1.00	0.91	0.15	0.88	-0.41	0.80	0.50
$x_{NUC}$	0.83	-0.46	0.91	1.00	0.41	0.98	-0.42	0.86	0.60
$x_{AP}$	0.26	-0.26	0.15	0.41	1.00	0.40	-0.21	0.28	0.26
$x_{OAB}$	0.81	-0.46	0.88	0.98	0.40	1.00	-0.42	0.88	0.61
$x_{CS}$	-0.49	0.89	-0.41	-0.42	-0.21	-0.42	1.00	-0.43	-0.30
$x_{IAB}$	0.78	-0.45	0.80	0.86	0.28	0.88	-0.43	1.00	0.54
$x_{LAP}$	0.53	-0.32	0.50	0.60	0.26	0.61	-0.30	0.54	1.00

Table 4.6: Correlation, using Pearson's coefficients, of the binary variables found in table 4.4.

	$\theta_{HAB}$	$\theta_{PP}$	$\theta_{PIA}$	$\theta_{DF}$	$\theta_{RP}$	$\theta_{PFM}$
$\theta_{HAB}$	1.00	0.534	0.415	0.199	0.08	0.20
$\theta_{PP}$	0.534	1.00	-0.236	0.038	0.131	0.30
$\theta_{PIA}$	0.415	-0.236	1.00	0.252	-0.031	-0.056
$\theta_{DF}$	0.199	0.038	0.252	1.00	-0.014	-0.024
$\theta_{RP}$	0.08	0.131	-0.031	-0.014	1.00	0.181
$\theta_{PFM}$	0.20	0.30	-0.056	-0.024	0.181	1.00

Table 4.5 show that  $x_{HAB}$ ,  $x_{IAB}$  and  $x_{OAB}$  are indeed heavily correlated with correlation of 0.78 or higher. In fact, five variables have a correlation of 0.7 or higher with the most important variable  $x_{HAB}$ . The binary variables, as seen in table 4.6, are not as correlated with 0.535 as the highest correlation coefficient between  $\theta_{HAB}$  and  $\theta_{PP}$ .

## 4.2.2 SelectFromModel

SelectFromModel is a method by scikit learn to extract the most important features [18]. The module was configured to select the 8 most important variables for two regressors, RF and linear

SVC. Specifically, the method works by passing a regressor which has an internal attribute which tracks the importance of each variable. The SelectFromModel method then selects the 8 most important variables, based on the internal importance attribute. The SelectFromModel offers several different measures for feature importance. In this report, the default measures were selected. For the random forest model, the default importance is decided by the Gini impurity. For the linear support vector classifier, the default importance is based on the l2-norm of the weight vector, where weights with a larger l2-norm gets a higher priority. The 8 highest ranked variables using RF and linear SVC are shown in table 4.7.

Table 4.7: The top 8 most important features, according to SelectFromModel using Random Forest (RF) regressor, as well as Linear Support Vector Classification (SVC) regressor.

Random Forest	Linear Vector Classification
$x_{IN}$	$x_{IN}$
$\theta_{RP}$	$x_{UC}$
$x_{NUC}$	$x_{GC}$
$x_{LAC}$	$\theta_{RP}$
$x_{GC}$	$x_{HAB}$
$x_{HAB}$	$x_{CPS}$
$x_{UC}$	$x_{NUC}$
$x_{AR}$	$x_{OAB}$

Notably, the two regressors agree somewhat on which picks are in the top 8, agreeing with 6 picks, although the order is different. Furthermore, by comparing with table 4.6, it can be noted that multiple variable picks are highly correlated and that the algorithms seemed to favour continuous variables.

### 4.2.3 Forward Stepwise Logistic Regression

Forward stepwise logistic regression is a greedy search algorithm. Specifically, it starts with a constant model and then iteratively fits all possible variables in the state space to the target variable. The variable with the highest score of a certain measure is kept in the model. Common measures used are for example log-likelihood function value, p-value or pseudo R-squared value. In this thesis, pseudo R-squared value is used as measure. In the next iteration step, the procedure is repeated with the constant and the best variable from the previous iteration. The procedure is stopped when 8 variables have been selected. The result of the forward stepwise regression is shown in table 4.8. The table further shows the pseudo R-squared value in each iteration step.

Table 4.8: The top 8 most important features, according to the forward stepwise logistic regression.

Variable	Pseudo R-squared
$x_{CPS}$	0.3154
$x_{HAB}$	0.4123
$x_{AR}$	0.4168
$x_{TR}$	0.4205
$x_{LAP}$	0.4244
$\theta_{PP}$	0.4272
$\theta_{RP}$	0.4292
$x_{CS}$	0.4312

Table 4.8 show that the pseudo R-squared value converges fast for the first two variables and then stagnates, not increasing much even if more variables are added. Moreover, the forward stepwise algorithm, by comparing variables in table 4.5, seemed to include less correlated variables compared to the SelectFromModel algorithms.

#### 4.2.4 Conclusions

Comparing the special features churn proximity score,  $x_{CPS}$ , and churn score,  $x_{CS}$ , the churn proximity score seem to be a better feature than churn score, as  $x_{CPS}$  ranks higher than  $x_{CS}$  in table 4.8. Moreover,  $x_{CS}$  is not present in table 4.7 whereas  $x_{CPS}$  is. Furthermore, as they have a Pearson correlation of 0.89 according to table 4.5, they are heavily correlated and  $x_{CS}$  should therefore be omitted as  $x_{CPS}$  is better.

Although different algorithms came to different conclusions on which features are the most important, they agree somewhat on which features are of interest. The combined knowledge from these algorithms assist in the modeling stage on which features to focus on. However, as the SelectFromModel method ranks the features based on the full state space, effects such as correlated variables and curse of dimensionality can affect the ranking of these methods. Curse of dimensionality describes the problem of exponential increase in volume associated with adding extra dimensions to Euclidian space, cf. [20].

### 4.3 Model A - Logistic Regression

Given the conclusion attained from the feature analysis, the model selected as logistic regression model is shown in equation 4.16.

$$p_i = \frac{e^{model_A}}{1 + e^{model_A}} \quad (4.16)$$

where;

$$model_A = \beta_0 + \beta_1 x_{CPS} + \beta_2 x_{TR} + \beta_3 \theta_{PP} + \beta_4 \theta_{HAB} \quad (4.17)$$

The labeled variables in eq. 4.17 are given in table 4.7 and table 4.2. The churn predictions are given by:

$$\hat{y}_i = \begin{cases} 1, & \text{if } p_i \geq 0.5 \\ 0, & \text{if } p_i < 0.5 \end{cases} \quad (4.18)$$

where 1 is interpreted as a predicted churned customer and 0 is interpreted as a predicted non-churned customer. The correlation for the continuous respective binary variables are shown in table 4.9.

Table 4.9: Correlations, computed with Pearson's coefficient, for the variables used in the logistic regression model.

Tabel A: Correlation of continuous variables in  $model_A$ .

	$x_{CPS}$	$x_{TR}$
$x_{CPS}$	1.000	-0.249
$x_{TR}$	-0.249	1.000

Tabel B: Correlation of binary variables in  $model_A$ .

	$\theta_{HAB}$	$\theta_{PP}$
$\theta_{HAB}$	1.000	0.534
$\theta_{PP}$	0.534	1.0000

The resulting predictions on the validation set respective test set, using measures defined in table 4.1, are shown in table 4.10. In the table, a naive predictor is used as a reference where the current churn state is predicted as the next churn state.

Table 4.10: Results on validation respective test set for the logistic regression, as defined by eq. 4.16.

Tabel A: Results on the validation set.

Measure	Naive Predictor - Validation Set	Model A - Validation Set
True Negative	46262	44824
False Positive	0	2119
False Negative	9003	2856
True Positive	0	6221
Accuracy	0.837	0.911
Precision	0.000	0.746
Sensitivity	0.000	0.685
Specificity	1.000	0.955
AUC	0.500	0.82

Tabel B: Results on the test set.

Measure	Naive Predictor - Test Set	Model A - Test Set
True Negative	30839	30063
False Positive	0	1277
False Negative	6138	1886
True Positive	0	4294
Accuracy	0.834	0.916
Precision	0.000	0.771
Sensitivity	0.000	0.695
Specificity	1.000	0.959
AUC	0.500	0.827

Table 4.10 suggests the model has a high accuracy of 0.911 on the validation set and 0.916 on the test set. However, as the data set is skewed with many active customers, this measure is deceiving as the naive predictor yields an accuracy of 0.837. Comparing the results of the validation set and test set, the logistic regression model showed stable results with similar results for accuracy, precision, sensitivity and specificity for the validation respective test set. Since one of the potential area of use of the model is targeted advertisement towards customers who are likely to churn, the sensitivity becomes an important measure. A high sensitivity means that the targeted advertisement is aimed at customer who are actually likely to churn. As this measure is the lowest, it suggest worse applicability of the model. Furthermore, as noted by the correlation matrices in table 4.9, the features  $\theta_{PP}$  and  $\theta_{HAB}$  are significantly correlated to each other, which is a key assumption of logistic regression. Consequently, this introduces uncertainties in the  $\beta$  - estimates and reduce the generalizability of the model. Note that the models are trained on customers who are currently non-churned and therefore the naive predictor never predicts a customer to churn.

## 4.4 Model B - Random Forest Model

The random forest model was made up of 1000 trees, each with a depth of 2. The random forest model performance was sensitive to the parameter choices where similar choices of parameters had vastly different performance results. The final parameter choices for the random forest model are shown in table 4.11.

Table 4.11: Parameters used in the random forest model.

Model B - Parameters used
$x_{AR}$
$\theta_{PP}$
$x_{LAP}$
$x_{TR}$
$x_{CPS}$
$x_{HAB}$

with the corresponding correlation matrix:

Table 4.12: Correlation, using Pearson's coefficients, of the continuous variables in model B found in table 4.11.

	$x_{AR}$	$x_{LAP}$	$x_{TR}$	$x_{CPS}$	$x_{HAB}$
$x_{AR}$	1.000	0.404	0.567	-0.352	0.612
$x_{LAP}$	0.404	1.000	0.276	-0.321	0.533
$x_{TR}$	0.567	0.276	1.000	-0.304	0.395
$x_{CPS}$	-0.352	-0.321	-0.304	1.000	-0.532
$x_{HAB}$	0.612	0.533	0.395	-0.532	1.000

The resulting prediction on the validation set respective test set is shown in table 4.13. In the table, the naive predictor and the logistic regression model (model A) are shown for comparison.

Table 4.13: Results on validation respective test set for the random forest model. The naive predictor and the logistic regression model (model A) are shown for comparison.

Tabel A: Results on the validation set.

Measure	Naive Predictor	Model A	Model B
True Negative	46262	44824	45303
False Positive	0	2119	1358
False Negative	9003	2856	3253
True Positive	0	6221	5570
Accuracy	0.837	0.911	0.917
Precision	0.000	0.746	0.809
Sensitivity	0.000	0.685	0.639
Specificity	1.000	0.955	0.971

Tabel B: Results on the test set.

Measure	Naive Predictor	Model A	Model B
True Negative	30839	30063	30438
False Positive	0	1277	401
False Negative	6138	1886	3047
True Positive	0	4294	3091
Accuracy	0.834	0.916	0.907
Precision	0.000	0.771	0.885
Sensitivity	0.000	0.695	0.504
Specificity	1.000	0.959	0.987

Table 4.13 suggests that the random forest model is less prone to classify customers as churned (TP+FP), compared to the logistic regression model. Moreover, the difference becomes more evident on the test set, as the random forest model only classified 3492 as churned, compared to 5571 of the logistic regression model. The large difference in performance on the validation-respective test set of the random forest model suggest the model does not generalize well to new data. Furthermore, the sensitivity measure is worse for the RF model, compared to the logistic regression model. The sensitivity is especially bad on the test set with only 0.504. Moreover, as table 4.12 show, the variables are correlated to each other and since the random forest model assumes the individual decision trees are uncorrelated, this can decrease performance. Furthermore, more rigorous testing of hyperparameters using an appropriate grid search and a more balanced data set could have further improved the performance of the RF model.

## 4.5 Model C - Support Vector Classifier

Compared to both the logistic regression model and the random forest model, the support vector classifier was the most robust w.r.t parameter choices as the model performance was not as sensitive to the choice of parameters. For the support vector classifier, the radial basis function kernel was applied and the hyperparameter C in eq. 4.10 was set to 1. The final parameters of



the support vector machine is shown in table 4.14.

Table 4.14: Parameters used for the Support Vector Classifier (SVC) model.

Model C - Parameters used
$x_{IN}$
$x_{UC}$
$x_{GC}$
$\theta_{RP}$
$x_{HAB}$
$x_{CPS}$
$x_{NUC}$
$x_{OAB}$

Table 4.15 show the results of the support vector classifier on the validation respective test set. The table also show the naive predictor, logistic regression model and random forest model for comparison.

Table 4.15: Results on validation respective test set for the support vector classifier (model C). The naive predictor, logistic regression model (model A), random forest model (model B) are shown for comparison.

Table A: Results on the validation set.

Measure	Naive Predictor	Model A	Model B	Model C
True Negative	46262	44824	45303	44632
False Positive	0	2119	1358	1771
False Negative	9003	2856	3253	2513
True Positive	0	6221	5570	6445
Accuracy	0.837	0.911	0.917	0.923
Precision	0.000	0.746	0.809	0.784
Sensitivity	0.000	0.685	0.639	0.719
Specificity	1.000	0.955	0.971	0.962

Table B: Results on the test set.

Measure	Naive Predictor	Model A	Model B	Model C
True Negative	30839	30063	30438	29433
False Positive	0	1277	401	1210
False Negative	6138	1886	3047	1784
True Positive	0	4294	3091	4296
Accuracy	0.834	0.916	0.907	0.918
Precision	0.000	0.771	0.885	0.780
Sensitivity	0.000	0.695	0.504	0.707
Specificity	1.000	0.959	0.987	0.961

Comparing the models in table 4.15, the support vector classifier and the logistic regression show similar results. Both models show stable results with similar metrics on the validation-respective test set. However, the support vector classifier had a slightly higher sensitivity compared to the logistic regression model, which is an important metric for churn models. Furthermore, the SVC model was less sensitive w.r.t parameter choices, compared to the logistic regression model. Moreover, the support vector classifier does not make any assumptions of uncorrelated features. The less strict requirement on correlated features is important as the features are significantly correlated.

## 4.6 Discussion & Conclusions

Given the performance metrics of table 4.15, combined with the observation that the features are significantly correlated, the support vector classifier performed the best. However, the result does not necessarily generalize well to general churn models as many factors affects the performance. For example, the choice of churn definition had a significant impact on the performance of the model. If the best performing support vector classification model was instead trained to predict customer with an altered churn definition of only customers who cancel their account as definition, as opposed to the current definition of customers who cancel their account or has not made any payment nor transactions within 90 days, the model performs significantly worse. This is natural, as predicting when a customer decides to cancel their account is a more difficult problem as each individual have an internal decision process when they make the commitment to cancel their account. As a consequence, the appropriate model choice and feature selection for a certain churn problem is highly affected by which churn definition is used, as the more difficult-to-predict churn definition would likely need a different set of features and more advanced algorithms to yield a sufficient performance.

Other factors could also affect which models is the most appropriate . For example, the best performing model might have been different had the features not been highly correlated. Moreover, the target distribution was skewed, with many more non-churned customers compared to churned customers. One simple way to account for the skewed distribution is to over sample the minority, the non-churned customer, and under sample the majority, the churned customer, on the model set to give a more even distribution. This might have led to overall better model performances and perhaps a different ranking of the best performing models. Furthermore, the support vector classifier is a computationally expensive algorithm, which might cause issues with larger data sets. Moreover, since SVC is a computationally expensive algorithm, it might not be optimal in ensemble models. Ensemble models takes into considerations the conclusions made from numerous models, cf. [21]. Ensemble models such as adaBoost have in turn shown great performance in classification problems.

To evaluate the usability of the best performing model, i.e. the SVC model, further investigation is required. Some proposals for further investigation could be to simulate the expected increase in revenue given the result metrics of the SVC model and given some necessary assumptions. Alternatively, the customer group could be randomly partitioned in two different subsets, one which have access to the churn model and one that does not have access to the churn model. If the churn model give usable insights, then the average revenue should be higher for the subset with access to the churn model. To further ensure stability of the different

algorithms, the validation- respective test set could be evaluated on different time instances. This would give more assurance that the model is resilient to seasonal trends and temporary fluctuations.

To conclude, the support vector classifier performed the best in this particular churn model, however, it is difficult to draw general conclusions comparing the performances of logistic regression, random forest model and support vector classifier for churn prediction applications, as many case-specific factors affect which algorithm performs the best.

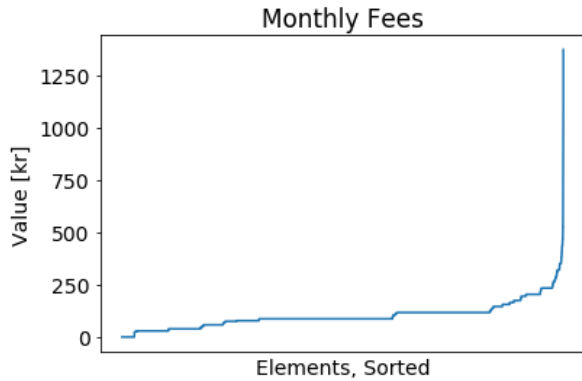
# Chapter 5

## CLV Model

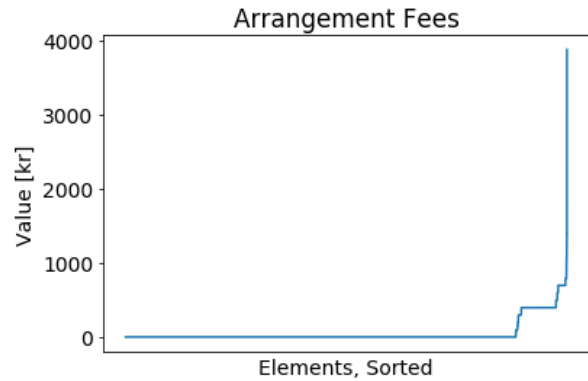
For the CLV model, two different algorithms will be used: A linear regression model and a support vector regression model.

### 5.1 CLV Data

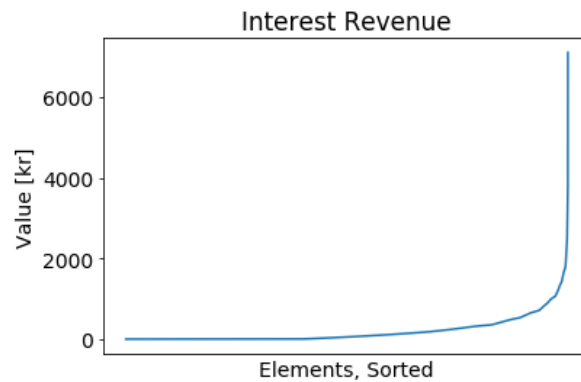
A schematic distribution, sorted by value, of respective revenue (arrangement fees, monthly fees and interest revenues) are shown in figure 5.1.



(a) Monthly fees sorted by value.



(b) Arrangement fees sorted by value.



(c) Interest revenues sorted by value.

Figure 5.1: Schematic distribution of monthly fees, arrangement fees and interest revenues, sorted by value.

Figure 5.1 show that monthly- and arrangement fees have a somewhat discrete distribution, whereas interest revenues has a more continuous distribution. This is because there is only a limited number of prices relating to different payment plans, yielding a more discrete character for the arrangement- and monthly fees. Interest revenues, on the other hand, depends on the value of the bought product, which can be any value. Furthermore, since the arrangement fees are paid only when initialising a payment plan, there are many zero valued arrangement fees.

Table 5.2 show the autocorrelation of respective revenue type for up to lag five. Figure 5.3 shows the original(reference) time series as well as the time series shifted by one quarter for respective revenue type. The data is sorted by the reference time series and to clarify the trend, a moving average of window size 50 have been applied to the shifted time series.

	$\tau = 0$	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$
<b>Interest Revenues</b>	1.00	0.90	0.81	0.74	0.69	0.64
<b>Arrangement Fees</b>	1.00	0.85	0.67	0.53	0.42	0.35
<b>Monthly Fees</b>	1.00	0.03	0.03	0.03	0.03	0.03

Figure 5.2: Autocorrelation of respective revenue type for lags up to five.

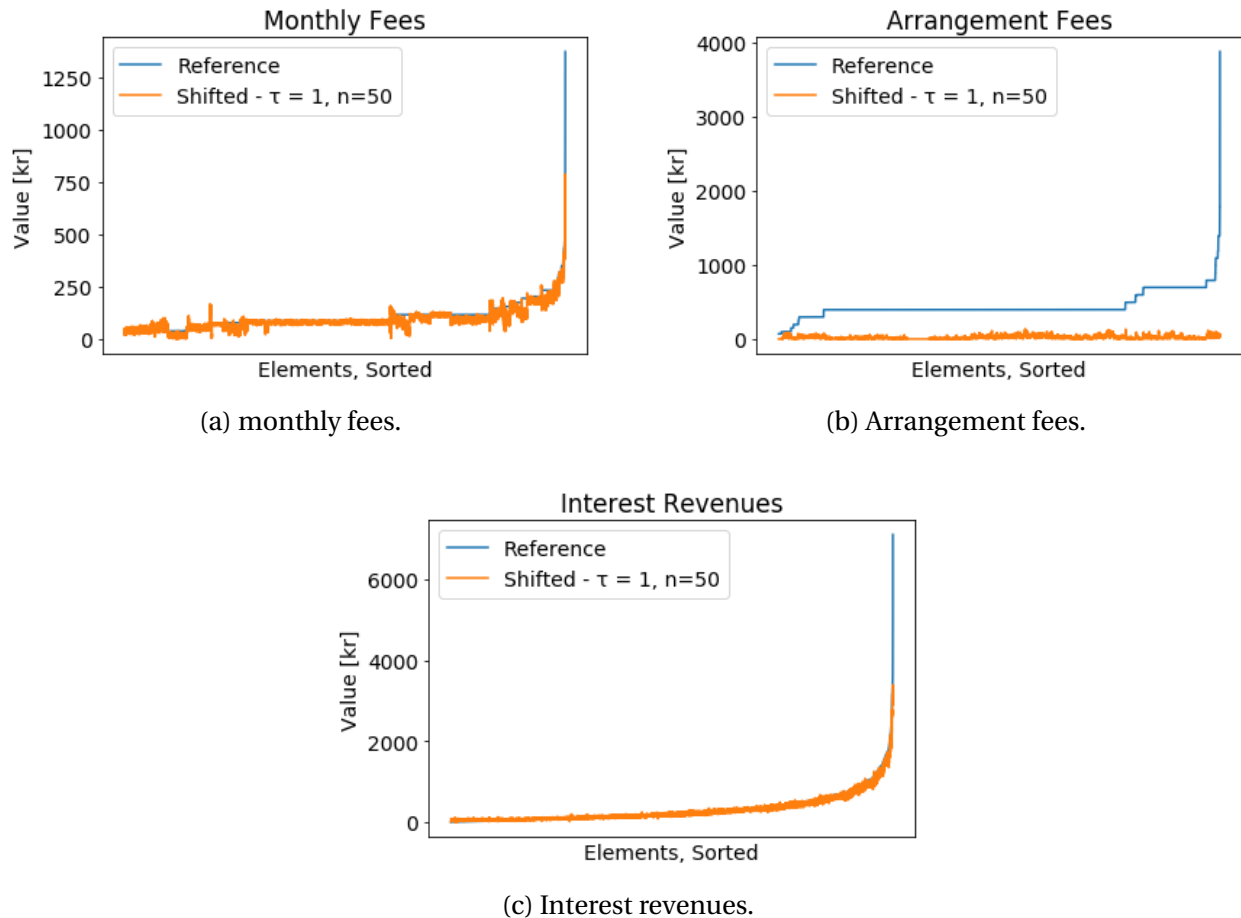


Figure 5.3: Original(reference) time series as well as the time series shifted by one quarter for respective revenue type. The data is sorted by the reference time series and a moving average have been applied to clarify the trend.

Figure 5.1 and figure 5.2 suggests that the monthly fees have a somewhat discrete distribution which has a high autocorrelation. This is due to the piecewise-constant nature of the monthly fees, where customers typically pay the same amount each quarter until either the payment plan is finished or the customer adds another payment plan. As the arrangement fees are paid only once per payment plan, the autocorrelation is close to zero, since an initial payment fee will typically not be paid two quarters in a row.

Since the arrangement fees are one-time events typically when the customer initialize their customer journey, it's not feasible to predict these, as there is no prior data at the start of the customer journey. Due to the piecewise constant nature of the monthly fees, a naive predictor, assuming it predicts the next quarter's revenue is the same as the current one, will predict the revenues exactly correct, unless a change in payment plan occurs. Such a change could for example be that the customer has finished paying a certain payment plan and the monthly fees drop to zero, or the customer adds a new plan and the value increases. Therefore, in order for a model to outperform a naive predictor, it is necessary with features that can indicate that an upcoming change might occur. However, as such features were not available, a model predicting monthly fees was omitted. The CLV model is therefore limited to predicting interest revenues.

## 5.2 Background

### 5.2.1 Linear Regression

Linear regression is a simple yet effective regression technique used in machine learning, cf. [22]. Linear regression assumes the data is described by:

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \dots + x_{ip}\beta_p + \epsilon_i \quad (5.1)$$

where  $y_i$  is the  $i$ :th data point,  $x_{ij}$  is the  $j$ :th feature of the  $i$ :th data point,  $\beta_j$  is the  $j$ :th coefficient corresponding to the  $j$ :th feature and  $\epsilon_i$  is normally distributed white noise with zero-mean and constant variance. Equation 5.1 can further be expressed more compactly on vector form:

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i \quad (5.2)$$

where;

$$\mathbf{x}_i = [1 \quad x_{i1} \quad x_{i2} \quad \dots \quad x_{ip}];$$

$$\boldsymbol{\beta}^T = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \dots \quad \beta_p]$$

Linear regression assumes the target variable is dependent of the features and that the features are independent, i.e.  $C[x_{ij}, x_{ik}] = 0 \quad \forall j \neq k$ . The regressor is given by:

$$\hat{y}_i = E[y_i] = E[\mathbf{x}_i\boldsymbol{\beta} + \epsilon_i] = \mathbf{x}_i\boldsymbol{\beta} \quad (5.3)$$

Using the regressor of eq. 5.3, the residual can be formed as:

$$r_i = y_i - \hat{y}_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i - \mathbf{x}_i\boldsymbol{\beta} = \epsilon_i \quad (5.4)$$

Equation 5.4 suggests that, if the deterministic trend of the data is found exactly, then the residual is expected to be zero-mean white noise with constant variance. It can be shown that the optimal estimation of the  $\beta$  coefficients in the regressor, eq. 5.3, is the solution to the least squares problem:

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad (5.5)$$

where  $\mathbf{y}$  is a vector containing the elements  $y_i$ ,  $\mathbf{X}$  is a matrix where the  $i$ :th row contains the features of the  $i$ :th element. One downside of linear regression is that it is sensitive to outliers,

as the minimization problem, eq. 5.5, includes a square operator, which will magnify the effects of outliers. Moreover, the constraints that the features need to be uncorrelated to each other is a strict constraint that is often not satisfied. Correlated feature variables will cause uncertainties in the  $\beta$  - estimates, leading to uncertain models. Furthermore, even if there exists a regressor which describe the trend of the data, it is not trivial how to find these features.

## 5.2.2 Support Vector Regression

Support vector regression is an extension to the support vector classifier, able to deal with regression problems. The idea of the support vector regression is to find a function  $f(x)$  which fit the observed data with at most  $\epsilon$  error from the function, cf. [23]. Similarly to support vector classifier, slack variables are added to allow some data points to deviate more than  $\epsilon$ , however, these points should be minimized. The function  $f(x)$  can typically be on any form but for simplicity, the function is assumed to be linear. i.e. of form given in equation 5.6.

$$f(x) = \mathbf{w}^T \mathbf{x}_i + b \quad (5.6)$$

with the linear function given by eq. 5.6, the support vector regression is found by solving:

$$\underset{\mathbf{w}}{\text{minimize}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (5.7)$$

subject to:

$$\begin{aligned} y_i - \mathbf{w}^T \mathbf{x}_i - b &\leq \epsilon + \xi_i \\ \mathbf{w}^T \mathbf{x}_i + b + y_i &\leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0 \end{aligned} \quad (5.8)$$

The minimization problem states that the linear function should be selected with a low slope as possible, while covering as many data points with at most  $\epsilon$  distance from  $f(x)$  as possible, allowing for some exceptions. This can be reformulated as  $\epsilon$ -insensitive loss function  $|\xi|$  given by:

$$\xi = \begin{cases} 0, & \text{if } \xi \leq \epsilon \\ |\xi| - \epsilon, & \text{otherwise} \end{cases} \quad (5.9)$$

A schematic figure illustrating the regression with slack variables  $\xi$  and  $\epsilon$  is shown in figure 5.4.



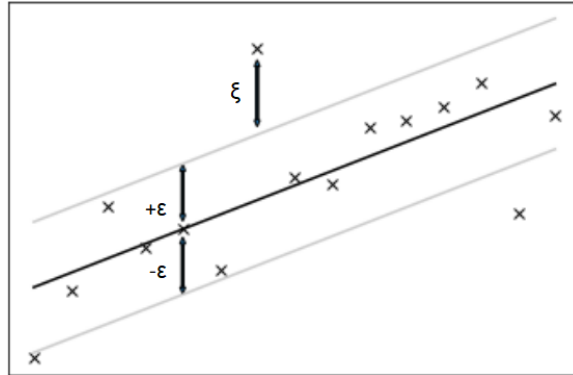


Figure 5.4: The figure illustrates the support vector regression algorithm and the slack variables  $\epsilon$  and  $\xi$  for a linear function, eq. 5.6.

The algorithm has two hyperparameters,  $\epsilon$  and  $C$ . For higher values for  $C$ , the algorithm will penalize outliers more, prioritizing the algorithm to avoid these to a greater extent. On the other hand, for lower values of  $C$ , the algorithm will instead prioritize selecting a function  $f(x)$  which is more flat.

### 5.2.3 Scikit Method - GridSearchCV

Many machine learning algorithms have one or many hyperparameters that needs to be tuned, e.g. support vector regression which has two. A popular method to perform hyperparameter tuning is the scikit method GridSearchCV [18]. The method exhaustively iterates through a parameter grid and performs a  $k$ -fold cross validation scheme for each parameter choice and selects the parameters which performed the best with regard to a certain score metric. A  $k$ -fold cross validation scheme partitions the data into  $k$  equally large subsets, where the model is trained on  $k-1$  subsets and evaluated on the remaining subset, with regard to a score metric. The process is then repeated for the remaining  $k-1$  subset such that each subset is the validation set once. The performance scores is then averaged. An example of a score metric is the mean squared error (MSE), in which case the search method would iterate through the parameter grid and select the parameter choice which had the lowest average MSE in the  $k$ -fold validation scheme.

## 5.3 Model A - Linear Regression

As the distribution of the interest revenue is somewhat exponential, the data is transformed to log domain. The original as well as the transformed data is shown in figure 5.5.

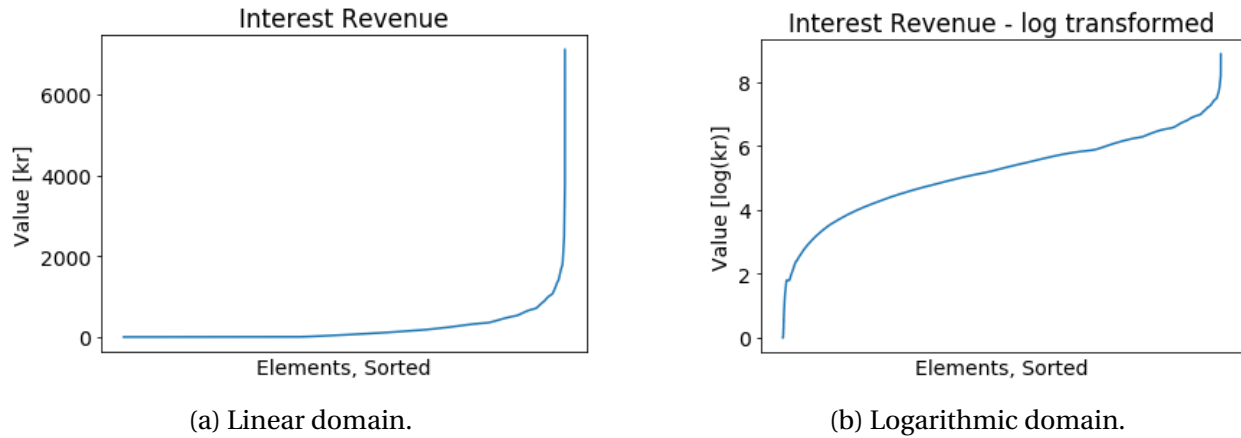


Figure 5.5: Schematic distribution of interest revenue in linear respective logarithmic domain.

Figure 5.5 show the interest revenue in log domain is closer to linear, which simplifies modelling. To find appropriate features, correlation between the target variable, i.e. interest revenue in logarithmic domain, and the feature variables is shown in table 5.1. In table 5.1, correlations between features are also shown.

Table 5.1: Correlations between target variable, log of interest revenues, and features. Also shows correlation between features. Correlations are computed using Pearson's coefficient.

Table A: Correlation between target variable and features.

	Target - $\log(x_{IR})$
$x_{UC}$	0.72
$x_{NUC}$	0.66
$x_{OAB}$	0.66
$x_{AP}$	0.65
$x_{IAB}$	0.59
$\log(x_{IR})$	0.59
$x_{HAB}$	0.45
$x_{IN}$	0.31
$x_{PPA}$	0.20

Table B: Correlation of features.

	$x_{UC}$	$x_{NUC}$	$x_{OAB}$	$x_{AP}$	$x_{IAB}$	$\log(x_{IR})$	$x_{HAB}$	$x_{IN}$	$x_{PPA}$
$x_{UC}$	1.00	0.90	0.88	0.71	0.80	0.48	0.75	0.08	0.14
$x_{NUC}$	0.90	1.00	0.98	0.78	0.89	0.58	0.74	0.26	0.21
$x_{OAB}$	0.88	0.98	1.00	0.79	0.91	0.59	0.73	0.11	0.15
$x_{AP}$	0.71	0.78	0.79	1.00	0.82	0.71	0.59	0.29	0.22
$x_{IAB}$	0.80	0.89	0.91	0.82	1.00	0.58	0.72	0.13	0.17
$\log(x_{IR})$	0.48	0.58	0.59	0.71	0.58	1.00	0.43	0.21	0.15
$x_{HAB}$	0.75	0.74	0.73	0.59	0.72	0.43	1.00	0.11	0.22
$x_{IN}$	0.08	0.26	0.11	0.29	0.13	0.21	0.11	1.00	0.08
$x_{PPA}$	0.14	0.21	0.15	0.22	0.17	0.15	0.22	0.08	1.00

with insights attained from the correlations in table 5.1, combined with some trial and error, the final parameters of the interest revenue model were selected as:

$$\hat{y}_i = \mathbf{x}_i \boldsymbol{\beta} \quad (5.10)$$

where;

$$\mathbf{x}_i = [x_{UC} \quad \log(x_{IR}) \quad x_{IN} \quad x_{PPA}]$$

Figure 5.6 show the prediction of the linear regression model, compared with a naive predictor. The naive predictor assumes the customer will generate the same interest revenues next quarter as the current quarter. In the figure, the left plot shows indices on the x-axis, whereas the right plot shows true revenue on the x-axis. In table 5.2, the mean error, mean squared error and variance of the residual, i.e. eq. 5.4, are shown. For comparison, the corresponding measures of the naive predictor are also shown.

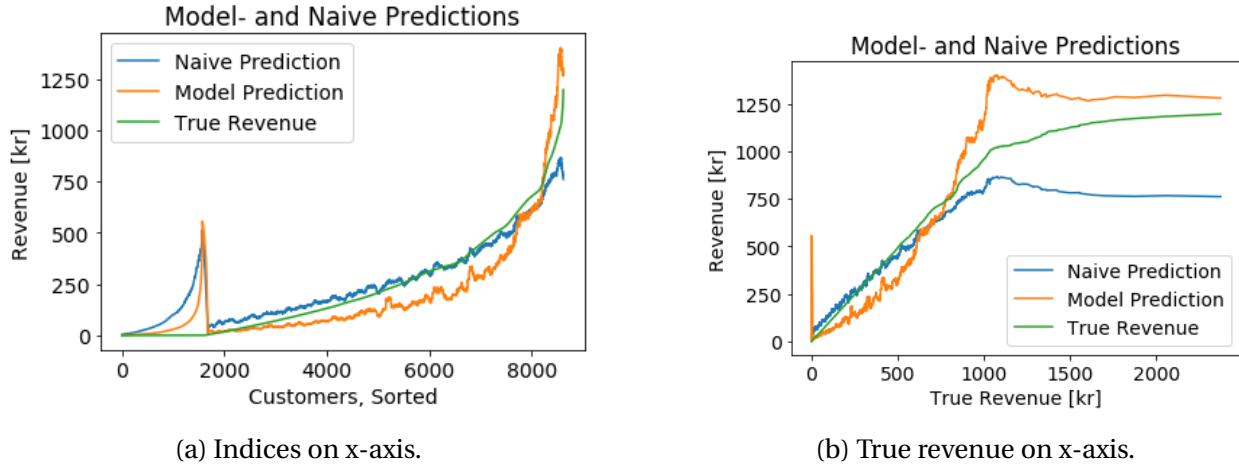


Figure 5.6: Linear regression model prediction respective naive model prediction of interest revenues. The plots are sorted by true revenue with two different values on the x-axis; indices respective true revenue. A moving average with window size 50 has been applied to clarify trend.

Table 5.2: Mean error, mean squared error and variance of residual of the linear regression model of the validation- respective test set. The corresponding values of a naive predictor, where current quarter's interest revenue is predicted for next quarter, are shown for comparison.

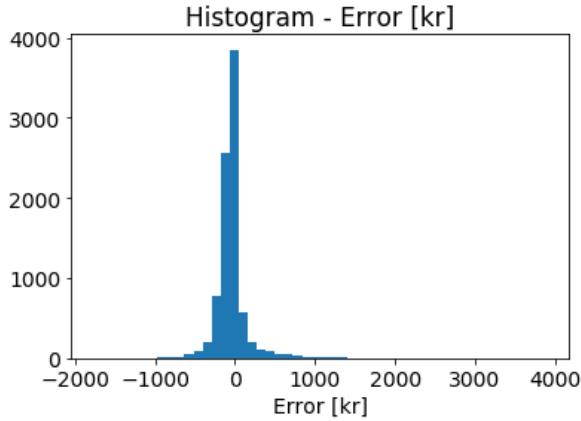
Table A: Results of validation set.

Measure	Naive Predictor	Model A
Mean Error	20.405	-41.588
Mean Squared Error(MSE)	25185	56178
Variance of Residual	24768	54448

Table B: Results of test set.

Measure	Naive Predictor	Model A
Mean Error	21.51	-38.02
Mean Squared Error(MSE)	22769	53011
Variance of Residual	22306	54448

Figure 5.6 suggests that the linear regression model undercompensates for customers in the mid-spending region while overcompensating for high generating customers. The linear regression model is better at predicting which customers will stop generating revenues next quarter, lower values in figure 5.6a. However, table 5.2 suggests the overall performance is much worse for the linear regression model for both the validation set and the test set, compared to the naive predictor. This can further be visualized by analysing a histogram of the residual, eq. 5.4, and a plot of the residual as a function of true revenue, shown in figure 5.7. In table 5.3, kurtosis and skewness measure of the residual is shown.



(a) Histogram of residual.



(b) Residual as a function of true revenue.

Figure 5.7: Histogram of residual of the linear regression model, as well as the corresponding residual as a function of true revenue.

Table 5.3: Kurtosis and skewness of the residual of the linear regression model. The corresponding kurtosis and skewness of the naive predictor is also shown for comparison.

Measure	Naive Predictor	Model A
Skew	-1.2	4.4
Kurtosis	14	51.1

Figure 5.7 further show the issue of the model undershooting in the mid-range region and overshooting in the upper range region. This is likely due to the minimization of least squares tend to overcompensate for outliers. Consequently, the model have difficulties modelling both the high- and low- spending customers, i.e. the tails of figure 5.5, at the same time. Moreover, had the model accurately been able predict the underlying trend, the residual would be expected to be normal distributed, as shown in eq. 5.4. However, the histogram in figure 5.7a and the values of table 5.3 (a normal distribution should have skew 0 and kurtosis 3 [24]) suggests that the residual is far from normally distributed. The linear regression model has two main issues; firstly, the features are correlated, which the linear regression model assumes are uncorrelated. Secondly, linear regression have difficulties dealing with outliers. The issue of correlation could be dealt with regularization techniques such as LASSO or ridge regression, cf. [22]. However, to deal with the extremities of high- and low spending customers, a model which is less sensitive to outliers is required, such as the support vector regression algorithm.

## 5.4 Model B - Support Vector Regression

For the support vector regression model, the same features are used as the linear regression, i.e. the parameters given in table 5.4.

Table 5.4: Parameters used for the support vector regression model.

Model B - Parameters used
$x_{UC}$
$\log(x_{IR})$
$x_{IN}$
$x_{PPA}$

The hyperparameters  $C$  and  $\epsilon$  were selected using the scikit-learn method GridSearchCV, with the method parameters presented in table 5.5.

Table 5.5: GridSearchCV method parameters used to tune  $C$  and  $\epsilon$  parameters for the support vector regression model.

C - grid	$\{2^k   k \in [-4, 4], k \in \mathbb{Z}\}$
$\epsilon$ - grid	$\{2^k   k \in [-8, 0], k \in \mathbb{Z}\}$
Number of Folds	5
Score Metric	Mean Squared Error(MSE)

The resulting optimal  $C$  respective  $\epsilon$  parameter from the GridSearchCV method, with the method parameters presented in table 5.5, were;

$$\begin{cases} \epsilon_{opt} = 2^{-6} \\ C_{opt} = 1 \end{cases} \quad (5.11)$$

Figure 5.8 show the prediction of the support vector regression model, compared with a naive predictor. The naive predictor assumes the customer will generate the same interest revenues the next quarter as the current quarter. In the figure, the left plot shows indices on the x-axis, whereas the right plot shows true revenue on the x-axis. In table 5.6, the mean error, mean squared error and variance of the residual are shown for the validation- respective test set. For comparison, the corresponding measures of the naive predictor and the linear regression model, model A, are also shown. The residual as a function of true revenue is shown in figure 5.9.

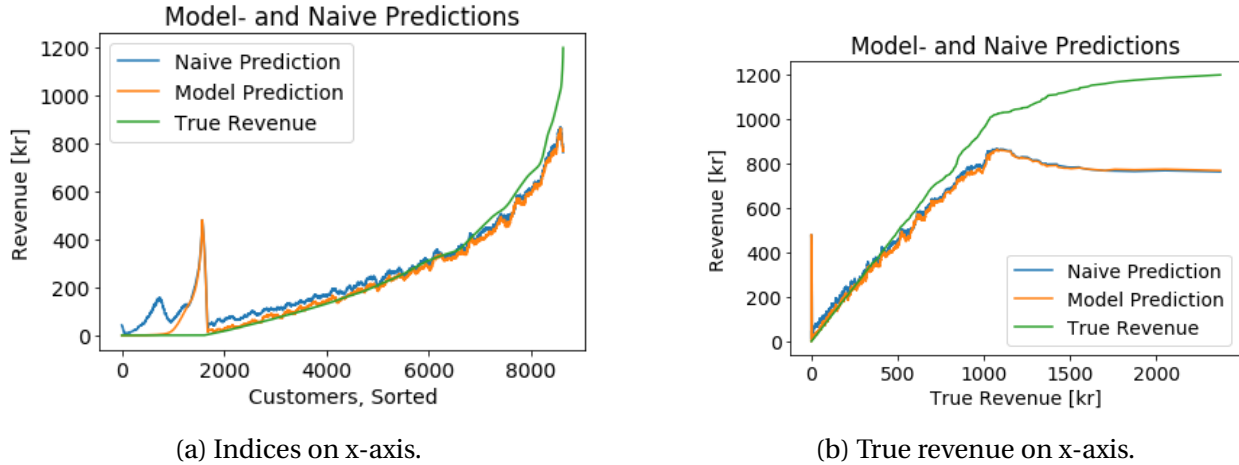


Figure 5.8: Prediction of the support vector regression model respective naive prediction of interest revenues. The plots are sorted by true revenue with two different values on the x-axis; indices respective true revenue. A moving average with window size 50 has been added to clarify trend.

Table 5.6: Mean error, mean squared error and variance of residual of the support vector regression model (model B) of the validation- respective test set. The corresponding values of a naive predictor, where current interest revenues are predicted for next quarter, and the linear regression model (model A) are shown for comparison.

Table A: Results of validation set.

Measure	Naive Predictor	Model A	Model B
Mean Error	20.405	-41.588	-8.756
Mean Squared Error(MSE)	25185	56178	22995
Variance of Residual	24768	54448	24769

Table B: Results of test set.

Measure	Naive Predictor	Model A	Model B
Mean Error	21.51	-38.02	-4.711
Mean Squared Error(MSE)	22769	53011	20165
Variance of Residual	22306	54448	20143



Figure 5.9: Residual as a function of true revenue for the support vector regression model.

Figure 5.8 show the support vector regression model performs significantly better than the naive predictor for all ranges except for the final 1/8:th highest generating customers, where the naive predictor is slightly better. Figure 5.9 further clarifies that the model is accurate for most of the customers but have a reduced accuracy for the highest generating customers. Interestingly, comparing the prediction of the linear regression model, figure 5.6, and the support vector regression model, figure 5.8, the linear regression model seem to predict customers in the lower ranges better. Table 5.6 further show that the support vector regression model, model B, performs better than both the linear regression model as well as the naive predictor.

## 5.5 Discussion & Conclusions

Comparing the two different algorithms, the linear regression was not appropriate to model this particular CLV data, as it performed significantly worse than the naive predictor. The support vector regressor outperformed the naive predictor across all ranges except the highest 1/8:th spending customers. Overall, the difficulty of predicting the interest revenues is that the distribution is skewed, with the majority of customers generating revenues less than 400 SEK and a smaller group generating significantly higher revenues. Regardless of which algorithm used, there will naturally be few data sample in the higher region to train the model on and, therefore, the accuracy will be lower. To further improve the model, more advanced ensemble models, i.e. models which take into consideration numerous weaker models, might improve the accuracy of customers in the low- respective high value regions, cf. [21]. An interesting example of model which could potentially improve the accuracy is the AdaBoost algorithm, which models the data in a sequential manner. After every model, the data is weighted based on how difficult the previous model had to predict the data, where difficult-to-predict data points, such as high or low spending customers, is assigned a higher weight. The next model in the sequential order will then focus more on predicting these difficult samples. Note that in the models in this report, customers who spent more than 5 standard deviations above the mean is treated as an outlier and was removed. Although these customers deviates significantly from the average, they likely should not be treated as outliers, as these customers are simply a high spending customer segment and would be interesting to model. However, in order to produce accurate models, it was necessary to remove them. Perhaps a boosting algorithm would be able to deal with these



customers as well. To further improve the CLV model, the data could be complemented with features which better predicts the pattern of a customer's monthly fees. With such additional data, a model which predicts monthly fees could be added. Moreover, the addition of cost data would enable a more complete model of a customer's CLV, one which includes cost.

The generalizability of the CLV model is difficult to assess, as the economic situation is in a period of fast change. For example, during the majority of the investigated period 2021Q1-2022Q2, the economy was in a boom with low interest rates and high spending patterns. After 2022Q2, the economy has stagnated significantly, with higher interest rates and a more defensive spending pattern. As a result, the models trained on low interest rates and a booming economy might not be particularly accurate today.

# Bibliography

- [1] Big data benefits: Study reveals increased revenues and reduced costs. <https://bi-survey.com/big-data-benefits>, Jul 2020. BARC. Accessed 2023/03/02.
- [2] Chittaranjan Andrade. Z scores, standard scores, and composite test scores explained. *Indian J Psychol Med*, 43(6):555–557, October 2021.
- [3] Zeljko Vujovic. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, Volume 12:599–606, 07 2021.
- [4] Vijay Kotu and Bala Deshpande. Chapter 4 - classification. In Vijay Kotu and Bala Deshpande, editors, *Data Science (Second Edition)*, pages 65–163. Morgan Kaufmann, second edition edition, 2019.
- [5] G. Tripepi, K.J. Jager, F.W. Dekker, and C. Zoccali. Linear and logistic regression analysis. *Kidney International*, 73(7):806–810, 2008.
- [6] Arun Addagatla. Maximum likelihood estimation in logistic regression. <https://arunaddagatla.medium.com/maximum-likelihood-estimation-in-logistic-regression-f86ff1627b67>, Apr 2021. Medium. Accessed: 2023/03/01.
- [7] OARC Stats. Faq:what are pseudo r-squareds? <https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/>. Accessed: 2023/03/01.
- [8] Wright MN Nembrini S, König IR. The revival of the gini importance? *Bioinformatics*, 34(21):3711–3718, 2018.
- [9] Ilyes Jenhani, Nahla Ben Amor, and Zied Elouedi. Decision trees as possibilistic classifiers. *International Journal of Approximate Reasoning*, 48(3):784–807, 2008. Special Section on Choquet Integration in honor of Gustave Choquet (1915–2006) and Special Section on Nonmonotonic and Uncertain Reasoning.
- [10] Wikimedia Commons. File:decision tree depth 2.png. [https://commons.wikimedia.org/wiki/File:Decision\\_Tree\\_Depth\\_2.png](https://commons.wikimedia.org/wiki/File:Decision_Tree_Depth_2.png), 2021. Online; accessed February 27, 2023. The picture is modified.
- [11] Asma’ Amro, Mousa Al-Akhras, Khalil El Hindi, Mohamed Habib, and Bayan Abu Shawar. Instance reduction for avoiding overfitting in decision trees. *Journal of Intelligent Systems*, 30(1):438–459, 2021.

- [12] Wikimedia Commons. File:random forest diagram complete.png. [https://commons.wikimedia.org/wiki/File:Random\\_forest\\_diagram\\_complete.png](https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png), 2017. Online; accessed February 27, 2023. The picture is modified.
- [13] Tony Yiu. Understanding random forest. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>, Sep 2021. Towards Data Science.
- [14] What is a support vector machine? *Nature Biotechnology*, 24:1565–1567, 2006.
- [15] Wikimedia Commons. File:support vector machines.png, 2020. Online; accessed February 27, 2023. The picture is modified.
- [16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
- [17] Wikimedia Commons. File:kernel machine.svg. [https://commons.wikimedia.org/wiki/File:Kernel\\_Machine.svg](https://commons.wikimedia.org/wiki/File:Kernel_Machine.svg). Online; accessed February 27, 2023.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [19] A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.
- [20] Eamonn Keogh and Abdullah Mueen. *Curse of Dimensionality*, pages 314–315. Springer US, Boston, MA, 2017.
- [21] Gavin Brown. *Ensemble Learning*, pages 312–320. Springer US, Boston, MA, 2010.
- [22] Johannes Lederer. *Linear Regression*, pages 37–79. Springer International Publishing, Cham, 2022.
- [23] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Aug 2004.
- [24] Measures of skewness and kurtosis. <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm#:~:text=Skewness%20is%20a%20measure%20of,relative%20to%20a%20normal%20distribution>. NIST.