DEVELOPING A LIFE INSURANCE RECOMMENDATION SYSTEM USING MACHINE LEARNING METHODS

ASLI HAZAL AKALTUN

JANUARY 2023

DEVELOPING A LIFE INSURANCE RECOMMENDATION SYSTEM USING MACHINE LEARNING METHODS

A THESIS SUBMITTED TO THE

GRADUATE SCHOOL

OF

BAHÇEŞEHİR UNIVERSITY

ASLI HAZAL AKALTUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF

BIG DATA ANALYTICS AND MANAGEMENT

JANUARY 2023

T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

MASTER THESIS APPROVAL FORM

Program Name:	BIG DATA ANALYTICS AND MANAGEMENT
Student's Name and Surname:	Aslı Hazal Akaltun
Name Of the Thesis:	Developing a Life Insurance Recommendation System using Machine Learning Methods
Thesis Defence Date:	23rd of JANUARY 2023

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Prof. Dr. Ahmet ÖNCÜ
Institute Director

This thesis was read by us, quality and content as a master's thesis has been seen and accepted as sufficient.

	Title/Name	**Signature**
Thesis Advisor's	Assoc. Prof. Tevfik Aytekin	
Member's	Prof. Dr. Songül Varlı	
Member's	Prof. Dr. Süreyya Akyüz	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Aslı Hazal Akaltun

Signature:

ABSTRACT

DEVELOPING A LIFE INSURANCE RECOMMENDATION SYSTEM USING MACHINE LEARNING METHODS

Aslı Hazal, Akaltun
Master's Program in Big Data Analytics and Management
Supervisor: Assoc. Prof. Tevfik Aytekin

January 2023, 95 pages

In the last 10 years, the use of advanced technologies and big data handling methods in the field of artificial intelligence has led to an increase in the number of machine learning-based projects in many sectors and domain such as personalized product offerings that enhance customer loyalty and business value. Algorithm based development has been ongoing for 70 years and continues to grow. The use of machine learning techniques in the insurance industry has the potential to greatly improve customer satisfaction and increase company profitability. In this study, by collecting and analyzing data on the portfolio movements, payment behavior, and demographic characteristics of existing product owners, predictive models were conducted to identify potential customers for cross-selling. This study followed data preprocessing steps, including handling missing data, detecting, and repairing outliers, and preprocessing categorical data for use in the model. The prediction problem was treated as a classification problem, and explanatory data analysis and correlation analysis were performed to gain a deeper understanding of the data. The results of this study could be used to inform future efforts to personalize product offerings and increase sales in the insurance industry.

The prediction problem was addressed using supervised learning methods, including Decision trees, Logistic regression, Random forest algorithms, Naive Bayes and Gradient boosting algorithms. The performance of the models was optimized through scenario-based experiments, and the effects of various data preprocessing steps, such as normalization and dimensionality reduction, on model performance were observed. The performance of the models was evaluated using a range of metrics, including accuracy, AUC, and F-1 scores. The results of this study suggest that hyperparameter Data Science, Data Mining, Recommender Systems, Machine learning, Supervised Learning tuning can play a significant role in improving the performance of machine learning models in this context. Overall, the use of machine learning techniques has the potential to greatly enhance the accuracy of predictions and improve decision-making in the insurance industry.

Key Words: Data Science, Data Mining, Recommender Systems, Machine learning, Supervised Learning.

ÖZ

MAKİNE ÖĞRENME YÖNTEMLERİ KULLANARAK HAYAT SİGORTASI ÖNERİ SİSTEMİ GELİŞTİRMESİ

Aslı Hazal, Akaltun
Büyük Veri Analitiği ve Yönetimi Yüksek Lisans Programı
Tez Danışmanı: Assoc. Prof. Tevfik Aytekin

Ocak 2023, 95 sayfa

70 yıl öncesinden günümüze dek gelişimi devam eden algoritmaların yapay zeka alanında ileri teknolojilerle ve büyük veri ile kullanılmasıyla son 10 yıldır kişiselleştirilmiş ürün teklifleri gibi birçok alanda müşteri memnuniyetini ve şirket karlılığını artıran makine öğrenmesi temelli projelerin sayısı her geçen gün artmaktadır. Hayat sigortası ve emeklilik sektöründe mevcutta ürün sahibi olan müşterilerin portföy hareketleri, ödeme davranışları ve demografik özelliklerinin verisi toplanarak ürününe sahip olan müşterilerin geçmiş verisi üzerinden gelecekte bu ürünü alma ihtimali olan müşterilere çapraz satış için tahminleme çalışması yapılmıştır. Bu çalışmada veri ön işleme adımlarından kayıp verilerin uygun şekilde kullanımı, aykırı değerlerin tespiti ve onarımı, kategorik verinin modelde çalışır hale getirilmesi için ön işleme, açıklayıcı veri analizi, değişilenler arasındaki korelasyon incelemesi gibi veri ön işleme ve veriyi tanıma adımları takip edilmiştir. Tahminleme problemi bir sınıflandırma problem olarak ele alınmıştır. Gözetimli öğrenme yöntemlerinden Karar Ağaçları, Lojistik Regresyon, Rastgele Orman Algoritması, Naïve Bayes ve Gradyan Artırma Algoritmaları ile tahminleme problem çözülmüştür.

Model performansını optimize etmek için senaryo bazlı deneyler yapılmıştır. Ön işleme, normalizasyon, çapraz doğrulama, boyut küçültme ve hiper-parametre ayarlama işlemlerinin model performansına etkisi gözlenmiştir. Doğruluk, AUC, F-1 skorlarına göre model performansları değerlendirmiştir. Hiper-parametre işlemlerinin model başarısında önemli derecede etkisi olduğu gözlenmiştir.

Anahtar Kelimeler: Veri Bilimi, Veri Madenciliği, Tavsiye Sistemleri, Makine öğrenmesi, Denetimli Öğrenme

To my mom and dad

ACKNOWLEDGEMENTS

I would like to thank my esteemed thesis advisor Assoc. Prof. Tevfik Aytekin
for his support and experience sharing and keeping me focused during the process.

To my mother Şükran, father Yavuz, brother Kemal, and lovely Ayas
I would like to express my heartfelt gratitude for the support and encouragement you have given me throughout my academic life. I would like to express my sincere gratitude to Prof. Dr. Yosum Kurtulmaz and Selçuk Yalçın for their support. I would like to thank Olcay Yöndem, Hazar Özyağcı, Yıldız Mine Kılıç, Erdem Işık and Başak Saban for their contributions.

.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

Chapter 1

Introduction

Digital transformation has become a crucial aspect of the modern economy, with AI-based applications playing a major role in this shift. According to Gartner's[1] 2023 technology trends report, the use of AI is expected to lead to a 50% improvement in customer loyalty and lifetime value, as well as business goals, by 2026. In addition, it is predicted that AI-based work will occupy 20% of the global workforce by this time (Top Strategic Technology Trends 2023, n.d.). Despite having been developed in the 1950s, machine learning and data mining algorithms and statistical knowledge continue to exhibit a growing applicability and significance in today's world. These technologies can aid organizations in decision-making processes and provide guidance for future development (Larson & Chang, 2016). As AI-based products and services continue to gain popularity, customer expectations are also evolving. Decision support systems aided by AI have been shown to have positive effects on customer retention and conversion rates when they are able to effectively analyze customer needs and expectations (Larson & Chang, 2016). The methods for analyzing customer expectations and priorities have evolved from creating rules based on business knowledge to using AI to uncover hidden knowledge in data and identify patterns. These methods are used in various industries, from e-commerce to insurance and pension, to create personalized and real-time outputs based on customers' historical data (Global Insurance Report 2023: Reimagining Life Insurance, n.d.). Machine learning algorithms are used to create decision support mechanisms in domains such as customer churn, sentiment analysis, demand forecasting, inventory management, fraud detection, price optimization, personalized customer service, market basket analysis, risk assessments, cross-selling campaigns, and customer underwriting. In this study, data mining and machine learning applications were used in a tailored and data-analytic study to predict personalized product offers for life insurance. Such personalized product offering practices, known as the "recommender system

[1] https://www.gartner.com/en

phenomenon," are commonly used in the e-commerce industry (Zhou et al., 2018). When a pattern is obtained in the data according to the past behaviors and demographic characteristics of the customers, the target customers who are likely to purchase the targeted product in the future can be predicted. These models in recommendation systems can provide the opportunity to filter by determining the target audience more clearly, thus reducing the costs of communication with the customer, as well as enabling an increase in sales as they offer the opportunity to reach the right customer audience. The product range in the e-commerce industry is quite extensive, and product offering models in this sector are highly advanced. In contrast, the product range and literature available on the life insurance sector are limited. Amazon[2], a leader in the e-commerce industry, offered product recommendations for 372 million products to its customers in 2017 using the "customer who bought this item also bought" feature (Zhou et al., 2018). In contrast, the number of products in the life insurance sector is in the low two digits. In this study, in order to capture actionable and robust patterns with customer data, comprehensive business understanding, data analysis, and pre-processing steps were followed, and then modeling was performed. After this stage, model evaluation was carried out. In this way, personalized customer needs and patterns are elicited. The advancement of data science is gaining momentum with various algorithms, particularly for personalized product recommendations. Collaborative filtering, content-based recommendation, recurrent neural networks, Boltzmann machines, CDL as a Bayesian generalized, and long-short-term memory (LSTM) models are some of the AI algorithms that are utilized for this purpose (Balabanović & Shoham, 1997; Pathak et al., 2010). When considering the quality and reliability of the data, it is observed that machine learning projects can only create business value through iterative and continuous improvement. It is important for every sector to observe a measurable business impact when conducting these studies (Larson & Chang, 2016).

[2] https://www.amazon.com/

1.1 Statement of the Problem

The aim of this academic study is to investigate the feasibility of utilizing machine learning algorithms and data mining techniques to accurately predict individuals who are likely to purchase life insurance products in the future, based on the historical data of current customers who have purchased pension products and who have target life insurance product from a company that sells life insurance and pension products. The motivation for this study stems from the desire to identify potential customers for cross-selling purposes within the company's portfolio. To address this research question, the study employs a combination of quantitative and qualitative methods, including the analysis of data collected from a sample of current customers and the implementation of various machine learning algorithms. The results of this study have the potential to provide insights for the company on the most effective strategies for targeting potential customers for life insurance products, ultimately leading to improved sales performance and customer satisfaction.

1.2 Theoretical Overview

Theoretical considerations of data pre-processing and evaluation in machine learning models have been investigated in order to improve the performance and accuracy of these models.

1.2.1 Missing values. It is important for model generalization that data analysis studies produce solutions for missing data. Also, many algorithms do not work when there is negative data. Standard regression programs can only work with data with complete information. On the other hand, Bayesian networks and tree-based algorithms have been examined as less sensitive to data occupancy. There can be many reasons for missing values in the data. One of them may be that the data provider did not record it correctly or that the customer or data owner did not share this information. As a solution for missing data, as a first step, the percentage of missing data in the total data is calculated. This stage is important for effect size (Cooper et.al., 1994). In the Missing value effect size analysis, the mean and standard deviation of all data, and the mean and SD of the filled information are calculated together with the missing and full data. As a method of handling missing values, deleting the relevant numbers if less than 10% of the data, filling the missing values with mean, and deleting the feature with more than 40% missing value can be applied. In addition, Buck’s method (imputing conditional mean) and the Maximum likelihood model are among the methods used. Although there are many methods, this subject still has a bias in the literature (Cooper et.al., 1994). Buck (1960) predicts by applying a regression model with an appropriate mean and covariance matrix. Buck's is not preferred because of the bias caused by imputed values. The maximum likelihood method has assumptions. Missing data is either not associated with any variable or has the same relationship with the observed variable. As a result, missing observations are ignored. KNIME data analytics platform provides a 'missing value' node to handle such problems.

1.2.2 Imbalance data. If there is a rare class as a modeling input in imbalanced datasets (Sun and Wong et. al., 2011), classification rules that small class is rare as well in learning algorithms such as Bayesian network, decision trees, and nearest neighbor. Resampling data space is one of the most used methods, there is a risk of overfitting when a small class is oversampled. Under-sampling can be applied for the prevalent class, there is a risk of losing information. Cost-sensitive boosting and cost-sensitive weighting data space are among the other methods used. However, extra learning costs and model overfitting problems can be observed. Another method is given as one-class learning. There are a limited number of learning algorithms for this data type. The 'Equal size sampling' node in the KNIME Data Analytics platform is used for imbalance situations. It has a SMOTE node that applies oversampling to the minority class using the Nearest Neighbour algorithm.

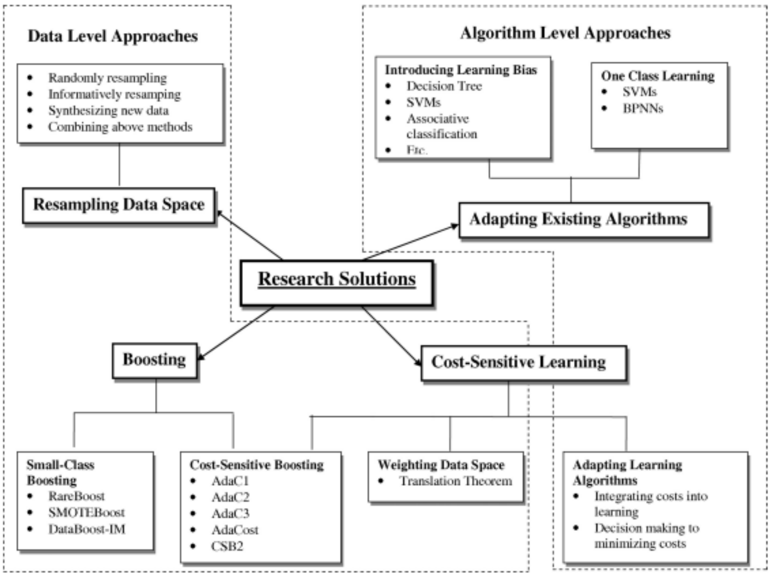


Figure 1. Handling imbalanced target variable

(Sun and Wong et. al., 2011)

1.2.3 Logistic regression. Logistic regression is a type of statistical analysis that is utilized to predict outcomes based on one or more independent variables. It can be used in various ways, such as predicting the probability of an event occurring, classifying data into categories, and determining relationships between variables. It uses a logit transformation on the odds which means the probability of success divided by the probability of failure. In data analysis and classification problems, regression models are used to find the relationship between the response variable and explanatory variables. In logistic regression applications, the dependent variable is taken as binary or dichotomous. The general logistic regression equation is introduced in the Figure below (Hosmer et al., 1989).

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Figure 2. Logistic regression

(Hosmer et al., 1989)

The maximum likelihood method plays a major role in setting up logistic regression. For using Logistic Regression Estimation, Maximum Likelihood focuses on maximizing the probabilities of obtaining observations. In logistic regression, the link function algebra illustrated in the Figure below (Hosmer et al., 1989) is a linear transformation of the dependent variable that yields a linear relationship with the independent variables. The slope coefficient in this model represents how much the logit changes for each unit change in an independent variable. Logistic regression assumes that observations are independent of each other and that there is no multicollinearity among the independent variables.

$$g(x) = \ln\{\pi(x)/[1 - \pi(x)]\} = \beta_0 + \beta_1 x$$

Figure 3. Logistic regression link function

Source: (Hosmer et al., 1989)

It is also important that the independent variables are linearly related to the log odds. This means that high correlations among features should be eliminated and the relationship between the independent variables and the log odds should be linear. To be able to evaluate model success, cost functions such as the sum of squared errors (SSE) can be minimized. Confusion matrix, F1 score and to be able to avoid overfitting cross-validation are other methods that can be utilized. Stochastic gradient descent (SGD) is an optimization algorithm utilized in logistic regression models that provides the minima or maxima of a given function. It does this by iteratively moving in the direction of the negative or positive gradient of the function, respectively. Unlike batch gradient descent, which computes the gradient using the entire dataset, SGD uses a randomly selected subset of the data to compute the gradient and update the model parameters in each iteration (Zou et al., 2019). This makes SGD faster and more efficient for training large and complex models. To be able to increase model performance, considering hyperparameter tuning the number of epochs can be adjusted. In the context of logistic regression, an epoch refers to a single iteration through the entire training dataset. While application of each epoch, the model parameters are updated considering the gradient of the loss function concerning the model parameters. A higher number of epochs can lead to better model performance. However, it can cause overfitting of the training data (Hosmer et al., 1989; Zou et al., 2019).

1.2.4 Gradient-Boosted tree. A gradient-boosted tree is a powerful machine learning technique that is utilized for classification and regression problems. As an ensemble method, which corresponds to the predictions of multiple sub-models to enhance the overall performance of the generalized model. The concept of gradient boosting was first introduced by Jerome H. Friedman (Friedman, 1999). In this paper, Friedman proposed using gradient descent to train a series of weak learners, which could then be combined to produce a strong overall model. One of the earliest boosting algorithms was developed by Robert Schapire, who introduced the AdaBoost algorithm in his 1995 paper "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting" (Schapire, 1995). The term "gradient-boosted tree" come into use later by Jerome H. Friedman, Trevor Hastie, and Robert Tibshirani, published in 2001(Friedman, 2001). Since gradient boosted tree as a key term utilizes weak learners such as a decision tree, minimizing the error between the predictions of this weak learner and the actual values of the training data are taken as the goal of the problem. For each subsequent weak learner, it is applied iterative (Friedman, 2001) processes. Once all of the weak learners have been trained, they are combined to produce the final gradient-boosted tree model. This is typically done using a weighted sum of the predictions of the individual weak learners, where the weights are determined by the performance of each learner (Krauss et al., 2017). The math behind gradient-boosted tree models is that training a series of weak learners, each of which is only slightly better than random guessing. It works by minimizing the error between the predicted values and the true values of the training data. This is done by iteratively updating the model parameters in a direction that reduces the error (Priyadarshini et al., 2019).

The gradient descent equation, which forms the basis of the gradient boosted tree algorithm, is given by: $\theta = \theta - \alpha * \nabla\theta\ J(\theta)$, where θ is the model parameters, α is the learning rate, and $\nabla\theta\ J(\theta)$ is the gradient of the cost function J concerning θ. The gradient-boosted tree model is created by combining the predictions of multiple weak learners. This is typically done using a weighted sum of the individual learners' predictions, where the weights are determined by the performance of each learner. The final prediction of the gradient-boosted tree model can be represented mathematically as: $y = \Sigma w * h(x)$, where y is the final prediction, w is the weight of each weak learner, and h(x) is the prediction made by each weak learner. This equation shows how the model uses the combined predictions of the weak learners to produce a more accurate final prediction. The process of adjusting the parameters of a machine learning model to optimize its performance is named Hyperparameter tuning. In the case of gradient-boosted tree models, several hyperparameters can be applied to enhance the performance of the model. In Gradient boosted tree models, it is adjusted the learning rate, the number of trees in the ensemble, the maximum depth of the trees, and the minimum number of samples required to make a split at each node as hyperparameter tuning. Grid search or random search is used as a tuning technique (Krauss et al., 2017). Grid search involves defining a grid of possible values for each hyperparameter and then training and evaluating the model for each combination of hyperparameter values. The random search involves sampling random values for each hyperparameter and then training and evaluating the model for each set of values. This process gives the opportunity to significantly improve the performance of the model.

2.1.5 Decision tree. Understanding intelligence as a phenomenon, machine learning algorithms have been developing since the 1950s. Decision trees demonstrate their importance in predictions through accomplished generalized models (Quinlan, 1986). Among diverse algorithms, in many disciplines, from statistics, and machine learning to pattern recognition, it is considered the most used approach. A decision tree is a structure composed of nodes, branches, and leaves that are built to model decisions and their probable outputs. As the components of the decision trees the nodes, branches, and leaves represent the following properties respectively, decision points, various paths, or options available to take from each node, and the result of each path. Their ability to visualize complex problems with multiple features increases make use of it in classification problems. From the roots of the decision tree to a leaf only subsets of the attributes are taken into consideration. When each internal node is divided into two and more sub-spaces regarding discrete input features. Probability of target value demonstrated via leaf. While decision trees help simplify complex data, they can sometimes pose difficulty with feature selection and hyperparameter tuning (Rokach & Maimon, 2006). On the other hand, it is regarded as much more comprehensible compared with other learning-based algorithms since stopping criteria allow controlling complexity. Methods of decision tree groups such as ID3 (Quinlan, 1986), CART (Classification and Regression Trees), and CHAID, are utilized to make variables discrete from the continuous variable form as well. Considering performance measurement as an evaluation criterion, minimizing the average node of a decision tree, as well as its depth, can be used as a success criterion for reducing generalization error. Lower depths and fewer nodes lead to fewer complex trees that are more likely to accurately predict unseen data. All research demonstrates that there is a trade-off between simplicity and accuracy criteria. To be able to solve this dilemma it is utilized pruning methods (Chou et al., 1989). Reduced error, minimum error, and cost-complexity pruning are some of the methods that can be applied. Originally pruning developed by Breiman et al., 1984, according to its findings, involves pruning after the tree has been fully grown by removing unnecessary branches from an already built decision tree model to reduce its complexity and improve its accuracy and generalization performance used (Rokach & Maimon, 2006). One of the significant pruning methods is Minimum Description Length (MDL) estimated by (Mehta et al., 1995):

$$\text{Cost (t)} = \sum_{c_i \in dom(y)} |\sigma_{y=c_i} S_t| . ln \frac{|S_t|}{|\sigma_{y=c_i} S_t|} + \frac{|dom(\mathrm{y})|-1}{2} ln \frac{|S_t|}{2} + ln \frac{\pi^{\frac{|dom(y)}{2}}}{\Gamma(\frac{|dom(y)|}{2})}$$

Figure 4. Minimum description length

Source: (Mehta et al., 1995)

Through the method, this method evaluates the complexity of a decision tree based on the amount of information needed to represent it by determining the amount of data needed to encode it. Regarding the advantages of decision trees on classification problems, it enables the following benefits. It is self-explanatory when it is applied a reasonable number of leaves comprehensible output on hand. Secondly, both nominal and numeric variables and outliers can be handled by decision trees. Moreover, it is capable of overcoming datasets with missing values. Since decision trees are the nonparametric method, they cannot encounter space distribution and classifier structure-related limitations. However, most of the algorithms, it is required a target variable with discrete properties and over-sensitive to the training set (Rokach & Maimon, 2006).

1.3 Purpose of The Study

The main academic aim of this study is to evaluate the factors that can impact the performance of a classification model in solving a problem in the field of life insurance. This research aims to address the limitations of existing approaches to product proposal and cross-selling in this industry by using advanced machine learning techniques to predict customer behavior and improve decision-making. By analyzing and comparing the results of different classification models, this study seeks to provide insights into the most effective methods for predicting customer response to life insurance products and maximizing cross-selling opportunities. The findings of this study have the potential to inform the development of more effective and efficient marketing strategies in the life insurance sector, ultimately contributing to the advancement of knowledge in this field.

1.4 Research Question

In this study, the life insurance product offering classification problem is addressed. The aim is to understand the extent to which factors influence the performance of the models and to identify the most effective approach for solving the classification problem.

Research Question 1: How it can be possible to develop a well-generalized product offering model for the life insurance sector using classification algorithms?

Research Question 2: "What is the effect of normalization, 10-fold cross-validation, standard train/test split method, feature selection, dimension reduction, outlier, and missing value processing, and hyperparameter optimization on the performance of machine learning models such as random forest, logistic regression, decision trees, naïve bayes, and gradient boosting?"

1.5 Significance of the Study

It has been determined in the literature research that studies are generally carried out in the field of e-commerce for a product offering. While the e-commerce product range is very wide, there are limited and complex product families in the insurance sector. This research is expected to provide valuable insights into the potential of similar customers to purchase whole life insurance and help the company develop effective strategies for cross-selling in the future by modeling the historical data of existing customers using supervised learning algorithms. These algorithms include decision trees, random forests, logistic regression, and XGBoost. With this approach, the goal is to conduct a cross-sell campaign targeting existing customers of the company in order to achieve the business goal. Additionally, the impact of data pre-processing and hyperparameter tuning on the model's performance will be investigated in a scenario-based analysis. The performance of the proposed model will be evaluated using metrics such as mean squared error, area under the curve (AUC), F-1 score, precision, recall, and accuracy. Comparing the success of modeling algorithms on different analytical platforms for machine learning will be the contribution of the study to the literature.

1.6 Limitations

Data leakage can occur when information from the training data set is used in combination with information from another source. This can result in overly optimistic performance on the test set, potentially leading to misleading conclusions. In real-world modeling, the use of information that is not available in the training data set can cause data leakage, resulting in models that perform poorly on unseen data despite achieving strong results on the test set. In the analysis of the data set, data leakage is likely to occur in the information on customer payment behavior. Economic changes can affect the accuracy of payment information obtained in the past, leading to a misalignment between the training and test data. To avoid data leakage, it is important to carefully pre-process the data and ensure that only relevant and representative information is used in the model. However, in this study, the customer's past account information was used. In addition, considering the practical application of machine learning and cloud-based data analytics come into prominence in this era. Since information in life insurance data is mostly confidential and restriction of personal data protection law, it is not allowed to utilize cloud technologies. Particularly, product offers in the e-commerce domain, cloud-based architectures are utilized to manage data from many different sources, increase data quality, manage information flow as close to real-time as possible and business value considering the characteristics of data which can be summarized as 5 “V”, that is, Volume, Variety, Value, Velocity, and Veracity. AI applications developed in the cloud regarding the elasticity and scalability to reduce costs adjust quickly to new changes and lead business dynamics. Leader cloud provides can be summarized as Alibaba, Amazon Web Services, Microsoft, and Google, their cloud-based architecture is scalable. Scalability is defined as the ability to handle an increased workload. Naive Bayes, support vector machine, logistics regression, and random forest, seasonal autoregressive integrated moving average (SARIMA), Gaussian White are algorithms that can be applied. Furthermore, many code base configurations are also possible. Amazon SageMaker is a cloud-based platform that facilitates the integration of machine learning into data analytics. It offers a range of tools and resources for building, training, and deploying ML models, including pre-built models and support for popular libraries and frameworks. With Amazon SageMaker, users can leverage powerful hardware resources to accelerate training times and easily deploy trained models to production environments. Overall, the platform simplifies the process of integrating ML into data analytics, making it more accessible and efficient for developers and data scientists. Amazon Simple Storage Service (S3) is a cloud-based data storage solution that enables users to store and retrieve data from anywhere on the internet. It is designed for large-scale data storage and web-scale computing. AWS Lambda is a serverless computing service that enables users to run code without the need to manage or provision servers. It is a cost-effective solution for running code for various applications and services (Yang et al., n.d.).

Chapter 2

Literature Review

2.1 Introduction

In literature research, machine learning algorithms and recommendation systems in the life insurance industry were investigated. The research has been conducted on Bayesian networks and Low-rank matrix factorization (LRMF) models, product recommendations in the insurance industry, supervised learning algorithms, data pre-processing steps, evaluation metrics, semi-supervised learning, and machine learning application tools (Qazi et al., 2017). The article utilized machine learning techniques to deliver value to both the customer and the company. The product recommendations obtained by machine learning indicate that customer satisfaction can be achieved by reducing the number of products and by detecting the product that is unique to the customer and the product that the customer is inclined to buy. The importance of such personalized product offerings was emphasized to build trust between customers and the company and to have loyal customers. Product offer studies for new customers in the insurance industry and for customers who already have a product in the company are the subject of this article. The portfolio is intended to offer personalized product offers based on similar customers. In this study, the customer information in the portfolio and the data expressing the customer's character are used as variables. According to the article, many missing data were encountered in the data. It has been found that the Bayesian network works well for such datasets (Qazi et al., 2017). Considering company profitability, the study focuses on the increase in customer retention, customer satisfaction, & customer lifetime value as the recommender system's ultimate goal for company profit. The scope of the study includes 3 states, Ohio (OH), Utah (UT), & Nebraska (NE). The reasons for preferring the Bayesian network for the products to be offered for up-sell and cross-sell are the low number of products, the missing values in the data, and the discrete data in the data set. Five different tools have been proposed for Bayesian network modeling. Since Bayesian lab has a wide range of feature sets, it has been the tool chosen for model development. It is also supported with java API for missing competencies. As model architecture and methodology, the dataset is split into a 70/30 train/test. When evaluating the success of the model, ROC curves, (Receiver Operating Characteristic) and AUCs (Area Under Curve) were used. Since the recommended number of products is few, considering the precision value for this model was found meaningless. One of the most important data pre-processing steps for a Bayesian network is that all data is discrete. Therefore, all continuous values were manually converted to discrete forms. Bayesian network is a probabilistic generative graphical model. Returns interest and conditional dependencies between variables via a direct acyclic graph (DAG). Since it works on local distribution and dependencies, it is a relatively more understandable model on the human side. Bayesian network modeling has been preferred since the prospective customer has a considerable amount of missing data. In this study, structured and unstructured learning methodologies are applied as follows. Initial step: starting point arc chosen as manually according to business logic. Tools used for BN learning: Weka BN tools, bn-learn for R, the Python library libpgm, Netica & BayesiaLab. BayesianLab has been chosen considering various feature sets.

Second step: running the Taboo algorithm. Third step: Augmented Markov blanket used for each target separately. The second step is regarded as surprising learning whereas Augmented Markov is defined as supervised learning. Concatenation of all networks without setting a target. “(3) Running various unsupervised algorithms from an empty starting point (i.e. only nodes, no added arcs). By allowing for more links to the target nodes and restricting links among other variables, we can apply various supervised learning algorithms to the data. This approach can help improve the accuracy and predictive power of the model by focusing on the most relevant features and relationships. By setting the local structural coefficients to 0, we allow the model to consider a wider range of possible connections between the target nodes and other variables. The model has been validated with offline and online tests. An observation period of 1 year was allowed to examine model success. The reason for this is that customer loyalty and retention rates, which are given as the purpose of the modeling, are metrics that can be evaluated over a long time. As a result, it has been obtained that business rules are very important and the BN approach performed better according to AUCs (Qazi et al., 2017).

Naika and Samant examined the performance of classification models using open-source data mining and modeling tools such as Knime, Weka, Rapidminer, Tanagra, and Orange. In this study, they aimed to examine the performance of various machine learning algorithms on a dataset of liver patient information. The dataset consisted of both numerical and categorical variables, and it is used to train and evaluate four different algorithms: decision trees, decision stumps, k-nearest neighbors, and Naive Bayes. They used a range of tools to analyze the data and build the models, including Weka, a widely used Java-based toolkit for machine learning applications, and Orange, a Python-based open-source tool developed by the Bioinformatics Laboratory of the Faculty of Computer and Information Science at the University of Ljubljana. In addition, it employed Tanagra, a tool for conducting parametric and nonparametric statistical tests, and Knime, a data analysis platform based on the Eclipse IDE. Their findings suggest that these algorithms can be effective in predicting liver disease status in patients, with decision trees and k-nearest neighbors achieving the best performance. The study revealed that the Naive Bayes algorithm had the lowest accuracy when implemented through WEKA. However, when utilizing the KNIME tool, the accuracy of the Naive Bayes algorithm was found to be higher in comparison to WEKA. Additionally, the decision tree algorithm also exhibited lower accuracy when implemented through WEKA, but again, the KNIME tool estimated a higher accuracy. In conclusion, the results indicate that the KNIME tool estimates a higher accuracy for all three classification algorithms when compared to WEKA. Further research is needed to confirm these results and to explore the potential use of these algorithms in clinical practice (Naik & Samant, 2016).

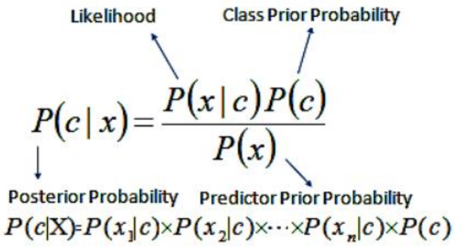
Considering the marketing domain, the study by (Villarroel Ordenes & Silipo, 2021) gives KNIME applications. It is visual-based programming. In the study emphasis on according to the Deloitte report, investments in machine learning-based digital marketing studies will increase by 20 percent in the next 3 years. Although it is predicted that the investments will increase, the expectations of the senior executives about the impact of these works on the business value are quite low due to the shortage of employees who can perform complex tasks. Programs such as SAS, Azure ML, R, KNIME, and ALTERYX were compared according to variables such as open source, visual programming, and open architecture (Villarroel Ordenes & Silipo, 2021). KNIME consists of workflows and nodes. Red marks are not configured, yellow configures, green executed, and red cross means error message. Data processing, image processing, text processing, and machine learning nodes such as unsupervised, supervised (as classification: decision trees, ensemble trees, logistic regression. as a numeric prediction: support vector machines and regressions. as both: neural networks and deep learning) and, semi-supervised exists. Data can be imported as databased or loaded in excel/CSV format if desired. Then, it is possible to obtain table join, column, row filtering, fold cross-validation, missing value operations, and prediction and scoring steps via nodes. It is also suitable for writing python scripts.

Bayes' theorem was discovered by mathematician Thomas Bayes in 1700. A Bayesian network is a probabilistic graphical model of conditional dependencies of a set of variables via a directed acyclic graph (DAG) reliance on Bayes' conditioning developed by Judea Pearl in 1985. Considering graphical models, nodes are a representation of variables and edges represent conditional dependencies (Murphy, 2006). In machine learning algorithms, conditional probabilities are estimated through training data. Based on the Bayesian probability inference, the conditional probability can be estimated from the statistical data and propagated along the links of the network structure to the target label (Mohanty et. Al., 2012).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 5. Bayes' Rule

The Naive Bayes algorithm is that Bayes Theorem-based classification technique. In this technique, mutual independence of the parameters (X_i) given a special variable (C or Y) is taken as the main assumption. Compared with Bayes' rule, it provides time and space advantages. Likelihood refers to the probability of observing data distribution given a specific situation in the data. Data distribution before making an inference represents prior. Updated probability of an event occurring after taking into consideration new information. Regarding conditional probability, it is mostly used for both classification and clustering machine learning problems.


Likelihood
Class Prior Probability
$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability
$P(c|X)=P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$

$$P\left(X_{1,\ldots\ldots,}X_n|Y\right) = \prod_{i=1}^{n} P(X_i|Y)$$

Figure 6. Naïve bayes

Source: (Murphy, 2006)

One of the most common problems when solving machine learning problems is using too many variables as model descriptors. Finding the variables that have a high impact on the prediction not only increases the success of the model but also helps to save time and space as it simplifies the complex structure of the model. Markov Blanket implies that nodes on a Directed Acyclic Graph can only be parent node dependent. The Parents do not take into consideration of information that comes from their ascendants, similarly, children cut information from descendants. The main difference between Markov Blanket and other supervised learning algorithms such as decision tress, SVM, and Decision trees is that it can eliminate redundant features. Bayesian networks (BNs) are probabilistic models that represent the joint distribution of a set of variables using a graph structure and associated conditional probabilities. The graph, which consists of nodes representing variables and edges representing relationships between variables, is subject to the Markov condition, which states that a node is independent of its non-descendants given its parents. The graph structure of a BN can be used to identify the minimal sufficient set of variables, known as the Markov blanket (MB), that shield a particular variable from the rest of the variables in the network. The conditional independence between two variables, X and T, given a set of variables Z, can be represented as I(X; T | Z) ≡ P(T | X, Z) = P(T | Z). The set of parents of a variable Xi in the BN is denoted as Pai (Tsamardinos et. al, 2003).

$$P(X)| = \prod_{\mathrm{i}=1}^{n} P_i(X_i|Pa_i)$$

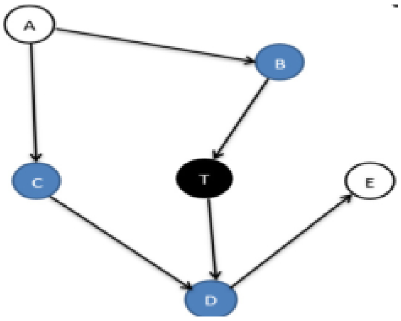


Figure 7. Markov blanket

(Mohanty et. al., 2012)

DAG Search, which is used to be able to handle complex decision problems, is a heuristic that enables a shortcut Tabu list name to prevent the search process from reverting to the previously visited solutions Tabu list name to prevent the search process from reverting to previously visited solutions (Mohanty et al., 2012).

$$TL = \{s^{-1}: s = \ s_i, i > k - t, \}$$

Figure 8. Tabu list

(Mohanty et al., 2012)

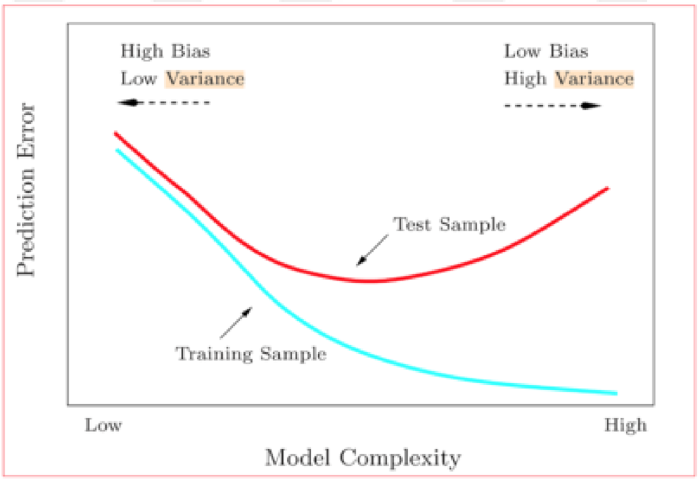


Figure 9. Model overfitting

(Nordhausen, 2009)

Chapter 3

Methodology

3.1 Research Design

This academic study has a methodology that consists of six main sections. In the first section, a literature review was conducted on product offerings and recommendation systems in the life insurance sector. Product offering studies were also examined in the e-commerce sector due to their widespread use in this industry. In the second step, the data preparation phase was initiated. Customer data from the company that sells life insurance and pension products were obtained for the dataset. The aim of the study is to create a machine learning-based modeling to predict customers who can have a tendency to buy cross-sell life insurance products. The dataset includes payment information for pension products and demographic data such as age, gender, and place of origin for the customer. The data collection process focuses on the customer's historical data for life insurance products in 2021. For negative sampling, a selection was made from the group of customers who have not had this product in their portfolio. In the literature review, it was examined that semi-supervised learning studies can also be conducted in cases where negative sampling is not clear. Academic studies found in literature as positive unlabelled learning were also examined (Bekker & Davis, 2018). In addition, academic research has been conducted on predictive models as a different discipline in product offering modelling with Bayesian networks Generative Adversarial Networks (GANs), collaborative filtering, content-based filtering, deep learning, long-short term memory (LSTM), low-rank matrix factorization, and tree-based algorithms have been observed in academic resource research for recommendation systems (Zhou et al., 2018). In the field of machine learning, there are four main categories of learning algorithms: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning algorithms are used to predict a certain output based on labeled input data. In the context of a classification problem, the target variable is divided into two or more class labels and the task is to predict the class label of a new example based on its input features. When applying supervised learning algorithms to a problem, it is common practice to divide the dataset into a training set and a test set. In the case of the product offer problem described in the research, the dependent variable is a binary variable indicating whether or not a customer has purchased the target life insurance product. This results in a classification problem with two class labels: (1)"owner and (0)"not the owner." Depending on the structure of the target variable, the classification problem may also be binary, multiclass, or multi-label. After conducting a literature review, the next step in the research process was to select a tool for implementing the machine learning models. Several options were considered, including the Python programming language, cloud-based platforms, and open-source analytical platforms such as KNIME Data Analytics, WEKA, and BayesianLAB (Naik & Samant, 2016). Based on the open-source nature, the range of applications, and the ease of use of these tools, it was decided to proceed with Python and KNIME Analytics for the modeling stage. In the fourth step, a detailed analysis of the obtained data was performed, including explanatory data analysis techniques such as descriptive statistics, histograms, and scatter plots. In the fifth step, machine learning modeling was carried out using Python and KNIME Analytics. The experiment setup was designed to test the effects of different machine learning algorithms and parameters such as normalization, forward feature selection, backward feature elimination, outlier handling, dataset split methods such as 10-fold-cross-validation, and hyperparameter tuning for Random Forest, Decision Trees, Logistic Regression, and Gradient Boosting. Finally, in the sixth step evaluation metric was used as illustrated on Figure 10.

Confusion Matrix	actual positive	actual negative
predicted positive	True positive (TP)	False positive (FP)
predicted negative	False negative (FN)	True negative (TN)

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN}$$

Figure 10. Confusion Matrix

Source:(Davis & Goadrich, n.d.)

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$

Figure 11. Model evaluation

During the literature review, it was found that various evaluation methods, such as ROC, F-1 score, mean squared error, and rank-based measures such as Mean Average Precision (MAP), Precision@K (P@K), and Cumulative Gain, have been used to assess the performance of various models (Yue et al., 2007). In the study, the model was trained on the training set and its performance was evaluated on the test set using various evaluation metrics, such as accuracy, precision, recall, F-1, and area under curve (AUC). The model with the highest performance on the test set is then chosen as the final model.

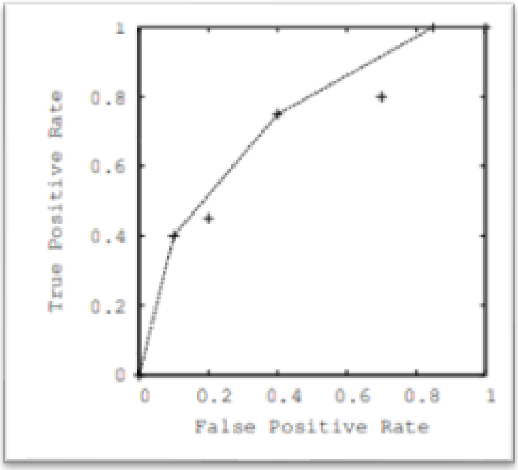


Figure 12. Model evaluation ROC curve - AUC

Source: (Davis & Goadrich, n.d.)

3.2 Dataset

The analysis of this dataset aims to predict customers who are likely to accept a cross-sell insurance product based on their demographic and behavioral data as well as their payment behavior for their existing pension products. The data used in this analysis was collected in 2021 and includes information on both pension product payment behavior and customer demographics. To ensure the validity of the predictions, the dataset includes a mix of customers who already own the insurance product and those who do not. The dataset was consisted of independent variables and 1 dependent variable as illustrated on Table 1 which included both continuous and categorical variables. To prepare the data for modeling, it is important to perform pre-processing steps to ensure that the chosen machine learning algorithms will work effectively and avoid overfitting or underfitting. One way to do this is by examining the data types using a platform such as Knime Data Analytics. This will allow for a better understanding of the data and help identify any potential issues that need to be addressed before modeling.

Table 1

Dataset Description

Variables	Description	Detail	Type
Customer_age	Customer Age	Age of people	Numeric
Customer_segment	Customer Segment: company has its own segmentation according to business rules	101: Elite plus 102: Elite 103: Special group 104: Standard group 105: Low 106: Lowest	Ordinal
Gender	Woman- Man	0-1	String
Owner_pension_insurance_fks	Special insurance Product for life and accidents	Count of products for individual customer	Numeric
owner_product_egp	Pension product after retirement	Count of products for individual customer	Numeric
ProductOwner_vasa_pension	A special group of products called as VASA	Count of product for individual customer	Numeric

Tablo 1 (cont.d)

Variables	Description	Detail	Type
ProductOwner_vasa_egp_pension	A special group of products called as VASA	Count of products for individual customer	Numeric
owner_iges	Pension product supported by employer	Count of product for individual customer	Numeric
owner_ggbes_pension	Pension product for companies	Count of product for individual customer	Numeric
Productowner_OKS_pension	Government incentive pension product	Count of product for individual customer	Numeric
ProductOwner_personal_accident_insurance	Special insurance Product for life and accidents	Count of product for individual customer	Numeric
ProductOwner_forchildren_pension	Special pension product for children	Count of product for individual customer	Numeric
Productowner_SKH_ insurance	Special insurance Product for life and accidents	Count of product for individual customer	Numeric
Productowner_forwoman_pension	Special pension product for woman	Count of product for individual customer	Numeric
Average_age_of_productlife_ forAll	Value indicates the number of days that the customer has been a client of the company	in day. Average for more than 1 product	Numeric
Age_of_oldest_product	Customer lifetime	in day for oldest product	Numeric
Age_of_newest_product	Customer lifetime	in day newest product	Numeric
Q1_paid_amount	First quarter paid amount	Continuous	Numeric
Q1_payableamount_due	the amount is written in the contract for the pension	Continuous	Numeric
Q1_rateof_paid	1stquarterpaid/due	0.0-1.0	Numeric
Q1_count_payable_due	amount written in contract for pension	Continuous	Numeric
Q2_paid_amount	Paid for the first quarter	Continuous	Numeric

Tablo 1 (cont.d)

Variables	Description	Detail	Type
Q2_payable_due	the amount is written in the contract for the pension		Numeric
Q2_rateof_paid	2. quarter rate	0.0-1.0	Numeric
Q2_count_payable_due	amount is written in the contract for pension	Continuous	Numeric
Q3_Quarter_paid_amount	3. quarter paid amount	Continuous	Numeric
Q3_payable_due	amount written in contract for pension	Continuous	Numeric
Q3_rateof_paid	3. quarter paid /due	0.0-1.0	Numeric
Q3_count_payable_due	amount written in contract for pension	Continuous	Numeric
Q4_paid_amount	Paid by customer	Continuous	Numeric
Q4_payable_due	amount written in contract for pension	Continuous	Numeric
Q4_rateof_paid	4. quarter paid /due	0.0-1.0	Numeric
Q4_count_payable_due	amount written in contract for pension	Continuous	Numeric
yearly_paid_amount	Total pain in current year	Continuous	Numeric
yearly_payable_due	amount written in contract for pension	Continuous	Numeric
yearly_rateof_paid	Paid/due	0.0-1.0	Numeric
yearly_count_payable	Numberof maturities	Continuous	Numeric
Q1_Extra_payment	pay in addition to their product plan to increase their asset	Continuous	Numeric
Q2_extra_paid_amount	Customer can pay in addition to their product plan to increase their asset	Continuous	Numeric
Q3_extra_paid_amount	pay in addition to their product plan to increase their asset	Continuous	Numeric
Q4_extra_paid_amount	pay in addition to their product plan to increase their asset	Continuous	Numeric
Q4_accountvalue_asset	Portfolio asset	4th quarter	Numeric
Yearly_extra_paid_amount	pay in addition to their product plan to increase their asset	Continuous	Numeric
Q3_ accountvalue_asset	Continuous	3rd quarter	Numeric

Tablo 1 (cont.d)

Variables	Description	Detail	Type
Q2_ accountvalue_asset	Continuous	2nd quarter	Numeric
Q1_ accountvalue_asset	Continuous	1st quarter	Numeric
Instant_accountvalue_asset	Continuous	Instant	Numeric
Paid_by_direct_debit	Payment Method	as percentage	Numeric
Paid_by_cash	Payment Method	as percentage	Numeric
paid_by_credit_card	Payment Method	as percentage	Numeric
Paid_by_bank_transfer	Payment Method	as percentage	Numeric
ProductOwner_group_pension	Group plans product type	Continuous	Numeric
PensionproductOwner_iges	Pension product supported by employer	Continuous	Numeric
aktif_tam_kaps_hayat	Target variable	0-1	Binary

Table 2

Distinct Values

Feature	Unique Count
Customer_age	93
Gender	2
Nationality	39
Occupancy	997
Customer_segment	6
Owner_pension_insurance_fks	15
Owner_product_egp	2
ProductOwner_vasa_pension	3
ProductOwner_vasa_egp_pension	2
owner_iges	6
Pension_std_group	8
ProductOwner_pension_for_youth	7
ProductOwner_katilim_std_personal_pension	5
Owner_pension_forWomen	8
ProductOwner_std_personal_pension	10
ProductOwner_oks	12
Pension_katilim_group	3
ProductOwner_pension_withoutgroup	15
PensionproductOwner_iges	3
Target	2

Chapter 4

Findings

Initially, considering explanatory data analysis, when analyzing the customer portfolio of the insurance company, it was found that the average age of customers is 36. The oldest customer is 94 years old. The customer segment variable is found to be an ordinal variable consisting of 6 categories. 106 corresponds to the lowest service level segment, while 101 represents Elite plus customers. According to the customer distribution before pre-processing, the densest cluster is observed to be 106. After the data pre-processing procedures, this group was identified as 105. Different evaluations have been examined according to the algorithms used, both before and after pre-processing. Customers of the insurance company can have various pension and life insurance products. These product types include ProductOwner_IGES, which corresponds to retirement plans paid for by the company on behalf of its employees working at various companies. The ProductOwner_OKS variable represents the retirement product that the customer is legally required to have in the country. In addition to the products owned by customers, the data set also includes variables such as the financial accumulation and payment method, frequency, and amount of the customer's retirement account. The target variable is the special life insurance product owned by customers, which is a life insurance product. A modeling baseline was established with 26,118 customers who have this product and 26,118 customers who do not. After data manipulation and data pre-processing steps, the dataset size has changed. It was observed that the payment behavior of the customers and their lifetime, which value indicates the number of days that the customer has been a client of the company are highly correlated. Before addressing the classification problem, missing value and outlier detection was performed on the dataset. The distribution of the positive and negative target variables was balanced in a way that would be suitable for training. To begin the process of solving the classification problem, the distributions within the histograms for continuous values were analyzed. Afterward, the process moved on to the implementation of five machine learning algorithms. The machine learning algorithms were developed by selecting the implementation of modeling by randomly choosing the negative set, which is one of the methods recommended by

Positive-Unlabelled Learning method for recommender systems (Bekker & Davis, 2018). AUC (area under curve) and accuracy score evaluation methods were selected. Gradient boosted tree gave the best result.

4.1 Data Types Conversion

The target variable was converted to a string or converted to binary format if any. Similarly, according to business information, variable types that the platform does not identify as appropriate were changed. For instance, the gender variable determined as numeric was selected as the nominal format. The amount of all data that was positive and unlabelled (or considered negative) is examined. It has been observed that there are 2.065.135 customers in total.

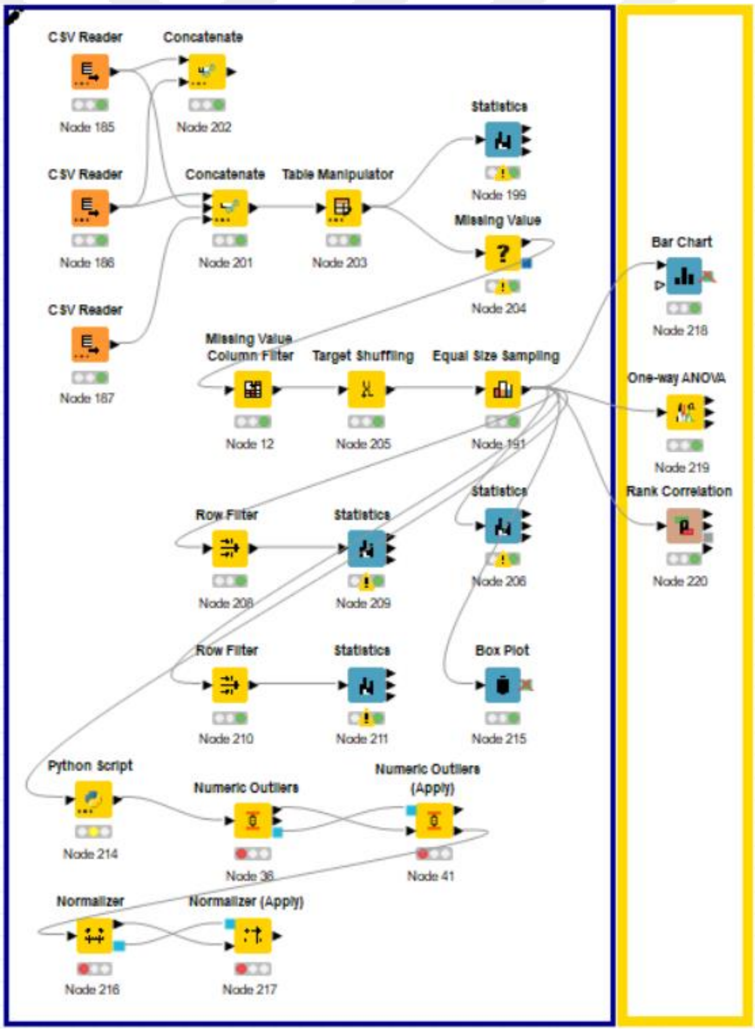


Figure 13. General data pre-processing architecture on KNIME Analytics

4.2 *Missing Value Operations*

According to the missing value analysis, if 40% of the column values are missing, these columns were removed. According to the data analysis, the hometown variable was omitted. Since Occupation and Segment variables have missing value rates of less than 25%, the most frequent value imputation is implied. Occupation variable was converted into dummy variable. Since it consisted of more than 900 unique variable, in order to mitigate the potential for bias in the dataset, it has been chosen to omit the occupation variable. This decision was based on the fact that a significant portion of the data was concentrated in a small number of instances. By removing this variable, it was aimed to reduce the influence of any potential biases that may be present in the data and ensure that our results are as unbiased as possible. It was observed that in the data set average age of the product, the oldest active product age and the newest product age parameters have missing values as 8.1%. Age, nationality, and hometown features also have a small number of missing values as shown in Table. Occupation has a 23.9% missing value considering the total amount of the dataset. Moreover over, the segment column also contains a very high rate of missing data with a rate of 11%. According to the literature review, it is known that missing value imputation can cause bias in the data set. Considering the risk of it, if the value is less than 10% the rows contain missing values omitted. The rows have a missing value for the age column, the average age of the product, oldest active product age, and newest product age column were omitted. Although hometown region variable has greater amount missing values, using hometown information it can be inferred regions through feature engineering.

Table 3

Missing Values Table

Average Age of Product	**Oldest Active Product Age**	**Newest Product Age**
8.1%	8.1%	8.1%
Age	**Hometown Region**	**Nationality**
0.0%	39.7%	0.0%
Hometown	**Occupation**	**Segment**
0.0%	23.9%	11.4%

In addition, since the hometown region has approximately 40% missing value, the column was omitted. Despite missing values on hometown region, subtraction can be performed from the hometown feature.

4.3 Imbalanced Target Variable

After the operation for the missing values in the data, the target variable sample sizes were resampled to be approximately the same. If one class dominates the other while modeling, the model will begin to predict the rarer group in the same way. To avoid this, it is important to organize imbalanced data sets. In addition, taking various random groups for the weighted negative set and including them in the modeling can contribute to the model's success. According to the literature review, under-sampling, oversampling, or SMOTE (Synthetic Minority Over-Sampling Technique) technique can be utilized to able have balanced data. Considering each technique has the possibility of causing overfitting or underfitting, under-sampling was applied, and the approximately same number of negative samples were taken randomly. Since pre-processing steps have been applied such as outlier and missing value elimination, the number of customers has varied.

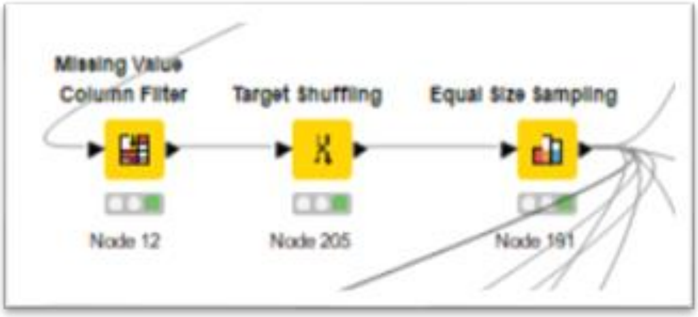
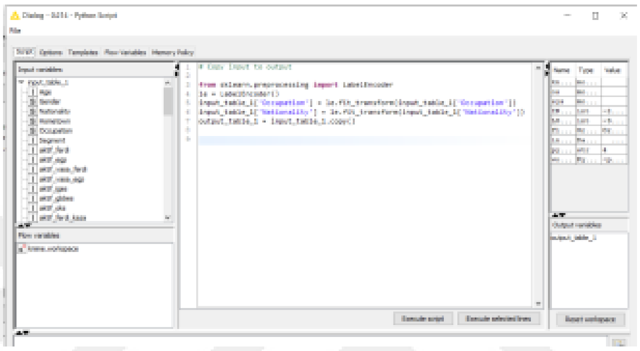


Figure 14. Equal-size sampling node

4.4 Handling Categorical Variables

On the Knime Analytics application, machine learning algorithms such as Support Vector Machine, Logistic Regression, and Decision Trees are required to convert all variables into the appropriate form. Categorical variables are transformed into dummy variables. In the literature review, it is observed that one-hot encoding, label encoding, and target encoding are the widely used strategies to handle categorical variables. In the insurance dataset, gender, nationality, hometown, and occupation variables were detected as a nominal value. In the data store, since the gender parameter was saved as a binary variable, only nationality and occupation parameters were converted into dummy variables by using label encoding. Since the diversity of each group is quite wide, it was preferred to use label encoding. Otherwise, it may cause a significant increase in dimension. Python script node on Knime utilized to apply to encode.



```
# Copy input to output

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
input_table_1['Occupation'] = le.fit_transform(input_table_1['Occupation'])
input_table_1['Nationality'] = le.fit_transform(input_table_1['Nationality'])
output_table_1 = input_table_1.copy()
```

Figure 15. Categorical variable operations

4.5 Handling Outliers

An analysis of outlier values in the data set revealed the presence of several continuous data with outlier values. In particular, the attributes of customer age, product age, customer lifetime, assets, and payables were identified as having outlier values. As a result, these outlying values were removed from the data set during the modeling process. It should be noted that methods such as replacing the outlying values with the mean or the most frequently occurring value were not utilized due to the potential for introducing bias into the data.

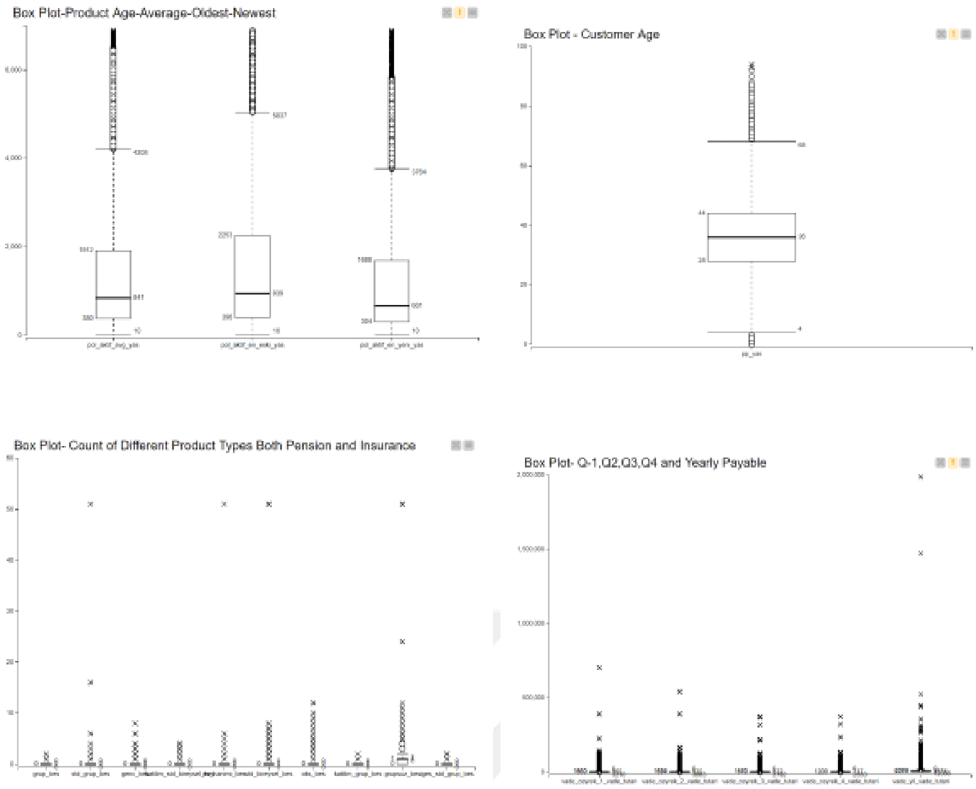


Figure 16. Handling outliers-box plot analysis

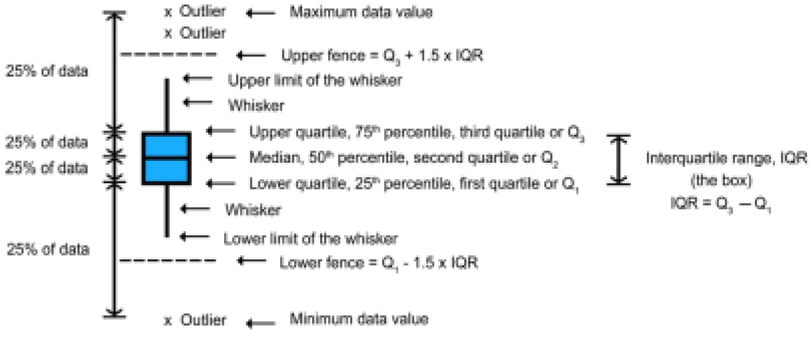

x Outlier
Maximum data value
x Outlier
Upper fence = Q_3 + 1.5 x IQR
25% of data
Upper limit of the whisker
Whisker
25% of data
Upper quartile, 75th percentile, third quartile or Q_3
25% of data
Median, 50th percentile, second quartile or Q_2
Lower quartile, 25th percentile, first quartile or Q_1
Interquartile range, IQR
(the box)
IQR = $Q_3 - Q_1$
Whisker
Lower limit of the whisker
25% of data
Lower fence = Q_1 - 1.5 x IQR
x Outlier
Minimum data value

Figure 17. Main components of a boxplot

Source:(v Ferreira et al., 2016)

4.6 Normalization

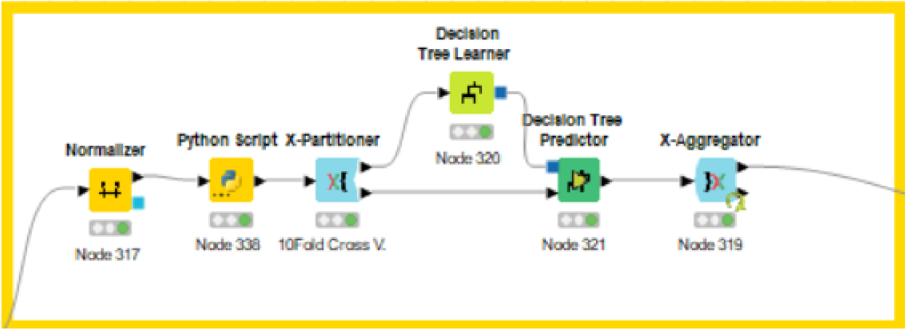


Figure 18. Normalization

Data mining is a process that helps to uncover hidden relationships within large sets of data. One of the techniques used in data mining is normalization, which has numerous methods available. al Shalabi et al. (2006) in their study conducted an experiment that compared the accuracy of different normalization methods, including Z-score, Min-Max, and Decimal Scaling. In their study, the normalized data sets were compared using the algorithms K-Nearest Neighbor (KNN), Local Transfer Function Classifiers (LTF-C), and decision trees. The results showed that Min-Max normalization method provided better accuracy. The normalization equations can be analyzed as follows (al Shalabi et al., 2006):

The Z-score normalization method utilizes the mean and standard deviation of the data to transform it. This method has the advantage of reducing outliers in the data.

$$x' = \frac{(x_i + \mu_i)}{\sigma_i}$$

Figure 19. Z-score normalization

On the other hand, Min-Max normalization shifts the features to a new range, often between 0 and 1 or between -1 and 1, preserving all relationships in the data.

$$x' = (x_{max} - x_{min}) \times \frac{(x_i - x_{min})}{(x_{max} - x_{min})} + x_{min}$$

Figure 20. Min-max normalization

Sigmoid normalization can be used if the parameters consist of noise data and similarly scales the data to a range of 0-1 or -1 and 1 (Jayalakshmi & Santhakumaran, 2011).

$$x' = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Figure 21. Sigmoid normalization

Decimal normalization, which is defined as decimal point movement based on the maximum absolute value of the attribute, is another method. Based on these findings, the Min-Max normalization method was selected, and the accuracy scores were evaluated considering machine learning algorithms. The logistic regression algorithm performed poorly without normalization. The performance of Naive Bayes, Random Forest, Gradient Boosted Tree, and Decision Tree algorithms was compared in terms of AUC, both with and without normalization applied to the life insurance dataset.

4.7 Descriptive Statistic

The data set consists of 95% continuous variables. According to the descriptive statistics analysis, it was observed that the standard deviation of many variables in the data was quite high. The Skewness values also show that the data does not follow a normal distribution. Positive Skewness corresponds to the tail on the right side of the distribution. It was determined that there was a significant variance in the average customer product age as customer lifetime.

Table 4

Descriptive Statistic

Variables	Min	Max	Mean	Std. Deviation	Variance	Skewness	Kurtosis
Customer_age	0	94	36.63	11.82	139.69	0.23	0.5
Customer_segment	101	106	104.46	1.43	2.06	(0.50)	-1.1
Owner_pension_insurance_fks	0	51	0.97	0.91	0.83	11.23	537.1
owner_product_egp	0	1	0.00	0.01	0.00	86.33	7451.9
ProductOwner_vasa_pension	0	2	0.00	0.02	0.00	65.79	4828.4
ProductOwner_vasa_egp_pension	0	1	0.00	0.00	0.00	228.46	52193.0
owner_iges	0	51	0.02	0.27	0.07	129.81	24083.7
owner_ggbes_pension	0	51	0.21	0.62	0.39	23.55	1732.9
Productowner_OKS_pension	0	12	0.14	0.43	0.19	4.86	49.2

Tablo 4 (cont.d)

Variables	Min	Max	Mean	Std. Deviation	Variance	Skewness	Kurtosis
Productowner_OKS_pension	0	12	0.14	0.43	0.19	4.86	49.2
ProductOwner_personal_accident_insurance	0	3	0.00	0.03	0.00	58.49	4406.1
ProductOwner_forchildren_pension	0	1	0.01	0.11	0.01	8.52	70.6
Productowner_SKH_ insurance	0	1	0.00	0.02	0.00	43.14	1859.2
Productowner_forwoman_pension	0	1	0.02	0.15	0.02	6.22	36.7
Average_age_of_productlife_forAll	10	6888	1336.1	1309.1	1713844.6	1.53	2.1
Age_of_oldest_product	10	6888	1453.4	1394.8	1945405.8	1.36	1.4
Age_of_newest_product	10	6888	1224.1	1315.4	1730196.4	1.66	2.4
Q1_paid_amount	0	701760	1375.4	4764.7	22702538.9	66.84	9042.5
Q1_payableamount_due	0	701752.35	1955.4	5911.9	34951070.1	43.72	4227.1
Q1_rateof_paid	0	6.73	0.6	0.5	0.2	(0.39)	-1.1
Q4_count_payable_due	0	306	3.4	3.0	8.9	28.63	2389.5
Q2_paid_amount	0	539580	1333.4	4179.6	17468882.4	48.69	5420.1
Q2_payable_due	0	539549.4	1782.2	5122.5	26239663.3	37.90	3095.1
Q2_rateof_paid	0	1	0.6	0.5	0.2	(0.52)	-1.7
Q2_count_payable_due	0	306	3.3	3.0	8.8	29.12	2458.1
Q3_Quarter_paid_amount	0	373014	1229.8	3922.4	15384926.1	28.80	1898.7
Q3_payable_due	0	372937.5	1649.3	5250.2	27565064.0	29.65	1602.9
Q3_rateof_paid	0	1	0.6	0.5	0.2	(0.47)	-1.7
Q3_count_payable_due	0	306	3.1	3.0	8.7	29.37	2502.0
Q4_paid_amount	0	373014	1019.9	3477.0	12089579.9	32.67	2674.8
Q4_payable_due	0	372937.5	1362.4	4429.2	19617732.6	28.90	1741.0
Q4_rateof_paid	0	1	0.6	0.5	0.2	(0.30)	-1.9

Tablo 4 (cont.d)

Variables	Min	Max	Mean	Std. Deviation	Variance	Skewness	Kurtosis
Q4_count_payable_due	0	306	2.9	3.0	9.0	28.02	2348.0
yearly_paid_amount	0	1987368	4958.5	13139.2	172637755.6	69.40	9982.4
yearly_payable_due	0	1987176.75	6749.3	16399.4	268939806.7	51.63	5363.3
yearly_rateof_paid	0	2.15	0.7	0.4	0.2	(0.68)	-1.4
yearly_count_payable	0	1224	12.7	11.6	134.5	31.10	2696.8
Q1_Extra_payment	0	400000	182.8	3318.9	11015237.4	59.17	5232.2
Q2_extra_paid_amount	0	250000	148.9	2519.3	6346971.0	50.47	3486.3
Q3_extra_paid_amount	0	800000	234.1	6025.5	36306852.7	77.63	7957.4
Q4_extra_paid_amount	0	407800	143.6	3571.0	12751856.6	66.19	5585.6
Yearly_extra_paid_amount	0	961400	709.4	9635.6	92844054.5	52.35	4070.0
Instant_accountvalue_asset	0	2.75E+07	34244.0	159848.0	25551379519.8	102.02	16850.6
Q1_accountvalue_asset	0	2.53E+07	30000.4	145409.7	21143987190.5	104.81	17509.2
Q2_accountvalue_asset	0	2.26E+07	25486.3	127719.4	16312251687.8	110.56	18928.4
Q3_accountvalue_asset	0	2.09E+07	22702.6	117053.8	13701588723.8	113.34	19599.7
Q4_accountvalue_asset	0	1.72E+07	16809.6	92803.9	8612568632.3	126.09	22819.4
Paid_by_direct_debit	0	1	0.23	0.40	0.16	1.31	-0.1
Paid_by_cash	0	1	0.13	0.30	0.09	2.25	3.6
paid_by_credit_card	0	1	0.54	0.47	0.22	(0.16)	-1.9
Paid_by_bank_transfer	0	1	0.02	0.13	0.02	6.98	48.6
ProductOwner_group_pension	0	2	0.00	0.04	0.00	28.73	912.0
Pension_std_group	0	51	0.06	0.36	0.13	57.13	7470.7
ProductOwner_pension_for_youth	0	8	0.16	0.41	0.17	3.13	15.4
ProductOwner_katilim_std_personal_pension	0	4	0.00	0.06	0.00	27.99	1043.8
Owner_pension_forWomen	0	51	0.08	0.38	0.14	50.29	6509.1
ProductOwner_std_personal_pension	0	51	0.16	0.50	0.25	22.45	1986.9
ProductOwner_oks	0	12	0.14	0.43	0.19	4.86	49.2
Pension_katilim_group	0	2	0.00	0.02	0.00	68.83	5321.7
ProductOwner_pension_withoutgroup	0	51	0.77	0.94	0.88	10.57	482.9
PensionproductOwner_iges	0	2	0.00	0.04	0.00	33.06	1224.4

Similarly, the paid amount variable was found to have a high variance for each quarter. To address issues of skewness and other problems with the distribution of this data, it was attempted to include the logarithm of the variables in our analysis.

However, this approach was not successful due to the creation of infinite variables in the Knime application. As a result, the logarithm of the variables was not included in the analysis. Through Kurtosis value it can be observed peak of frequency distribution. Most of the mentioned product counts had high Kurtosis value. Maximum yearly payment by customer detected as 1,987,368 TL and on average 4,958.5.

4.8 Explanatory Data Analysis

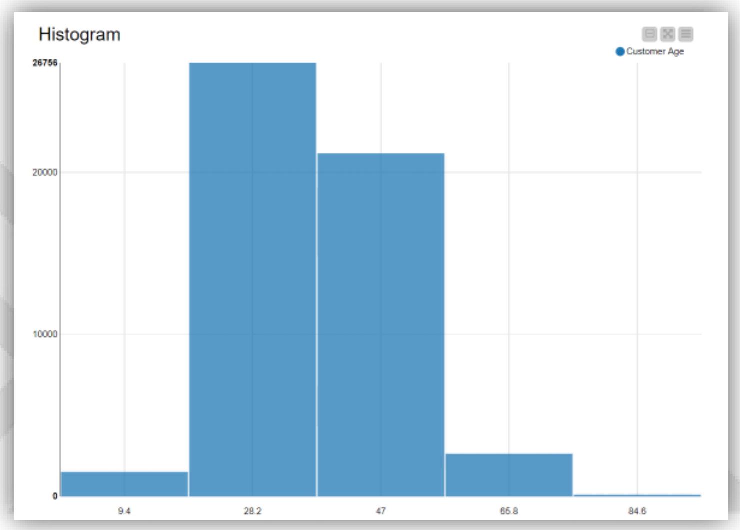


Figure 22. Customer age histogram

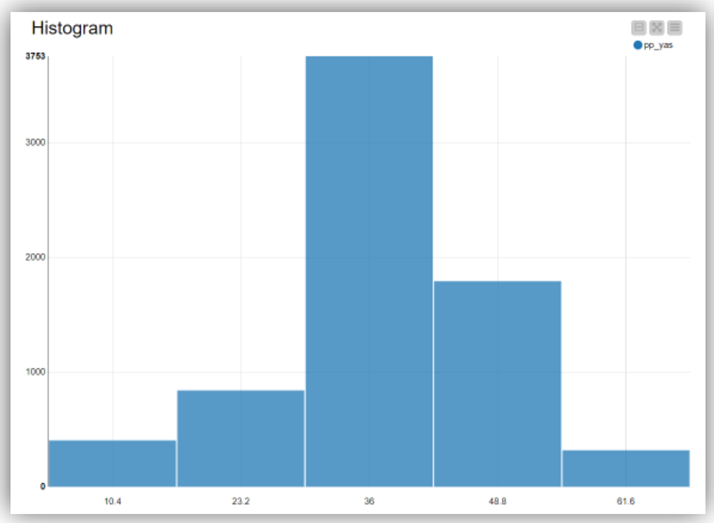


Figure 23. Customer age histogram after pre-processing

After pre-processing, the customer's age showed a concentration of 36 years old, while raw data showed a concentration at 28 years old.

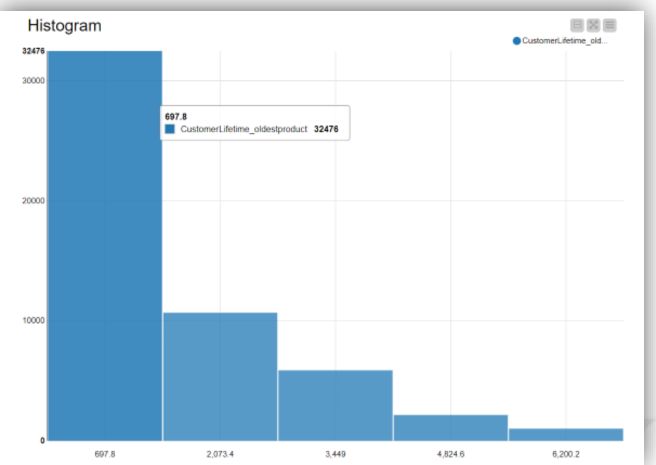


Figure 24. Customer lifetime histogram concerning the oldest product

The number of days the customer has been a customer of the company ranges approximately between 600 to 6,000 days, with the highest concentration around 600 days.

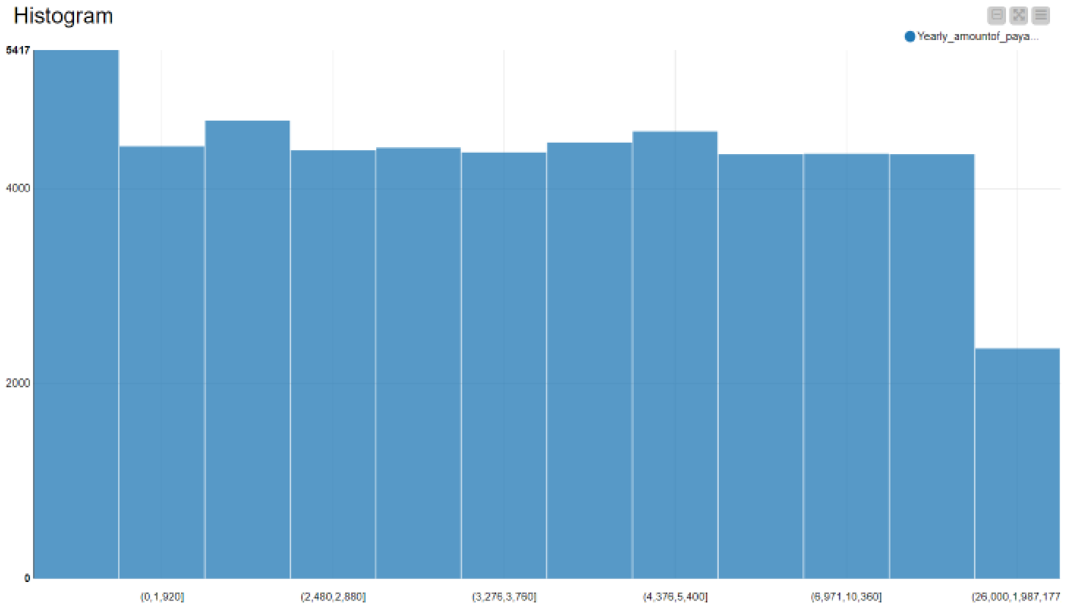


Figure 25. The yearly amount of payable(due)- frequency histogram

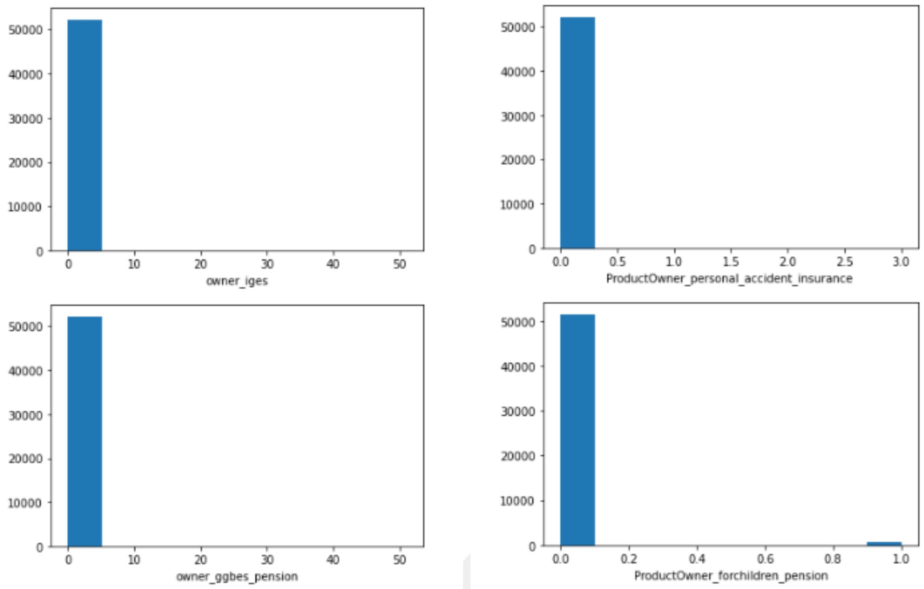


Figure 26. Different product group ownership

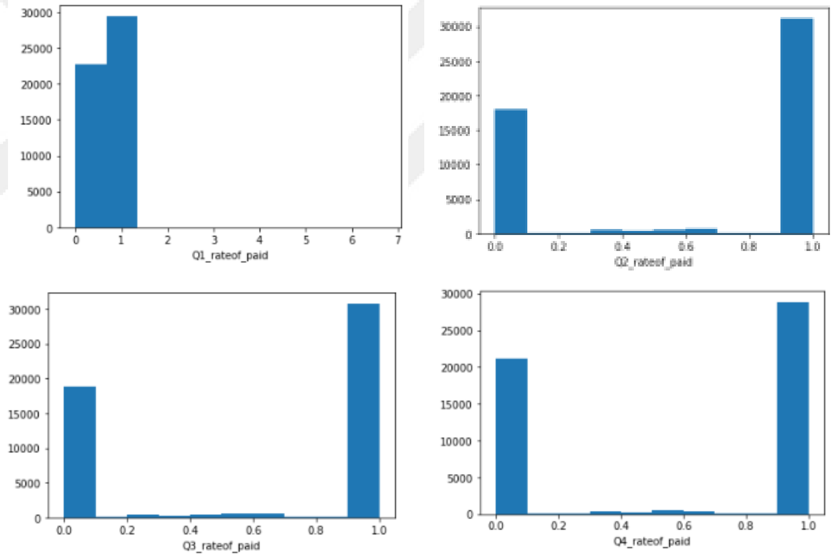


Figure 27. Quarterly paid rate

According to the customer's pension agreement, the amount to be paid regularly per period is specified. The payment rate has been evaluated for each quarter based on whether the customer makes or does not make this payment. Based on this rate, the customer either makes or does not make the payment.

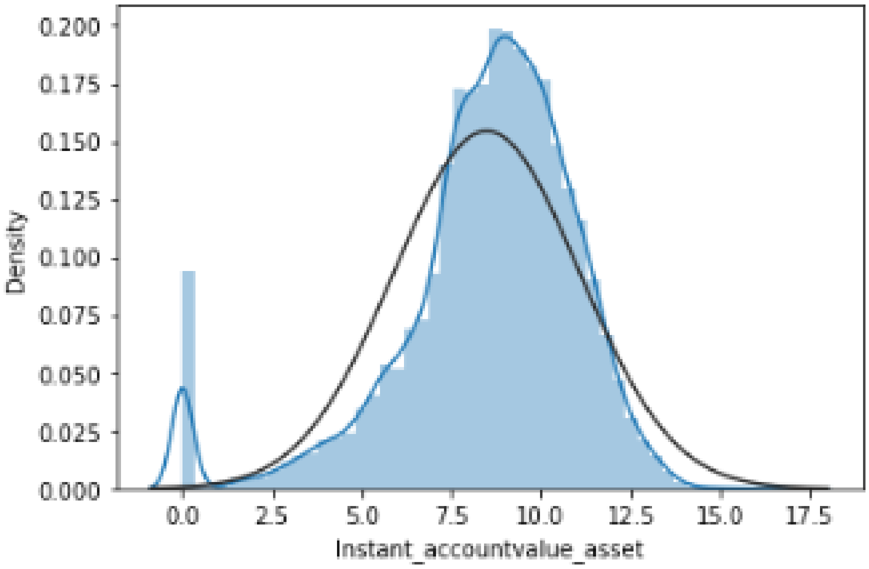


Figure 28. Instant account value

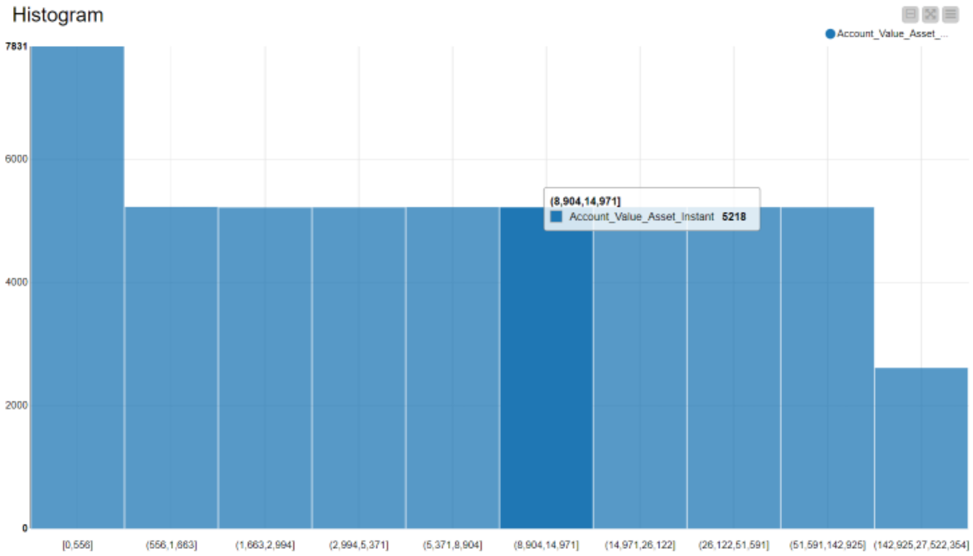


Figure 29. Instant account value (binning method= sample quantiles)

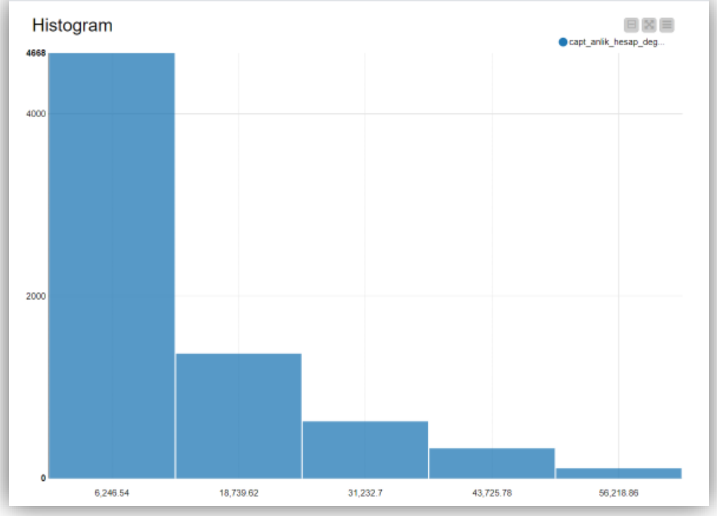


Figure 30. Instant account value after pre-processing

As customers make payments, a savings accumulation occurs in their accounts. These accumulations are valued in different funds. Based on these results, the distribution has been analyzed using different binning methods in histograms. The group excluding customers with no accumulation in their accounts can be considered to roughly have a normal distribution. Values in the approximate range of 0 to 27 million have been observed. After pre-processing, the customer's savings show a distribution in the range of approximately 6,000 to 56,000.

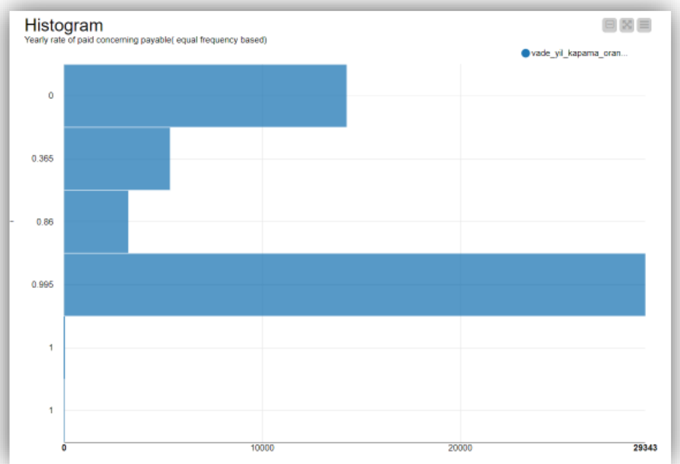


Figure 31. Yearly paid rate

Customers mostly make their payments according to the yearly paid rate.

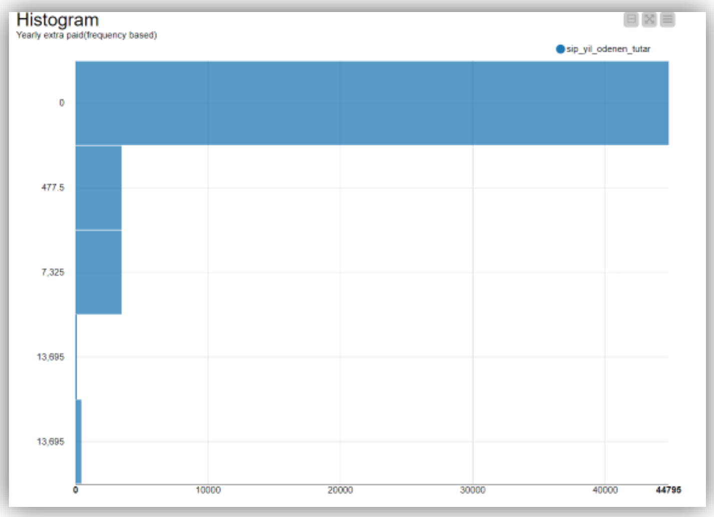


Figure 32. Yearly extra paid (frequency based)

The majority of customers do not make extra payments.

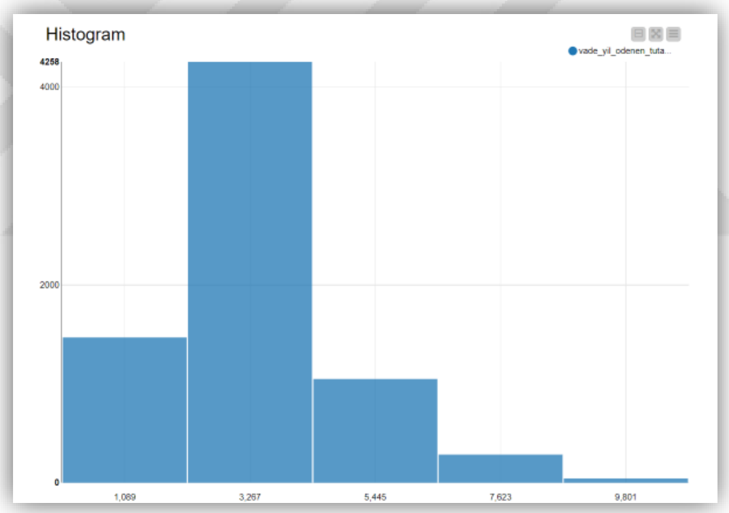


Figure 33. Yearly paid amount after pre-processing

After pre-processing, the distribution of maturities shows a concentration around the value of 3,000 in terms of annual payment.

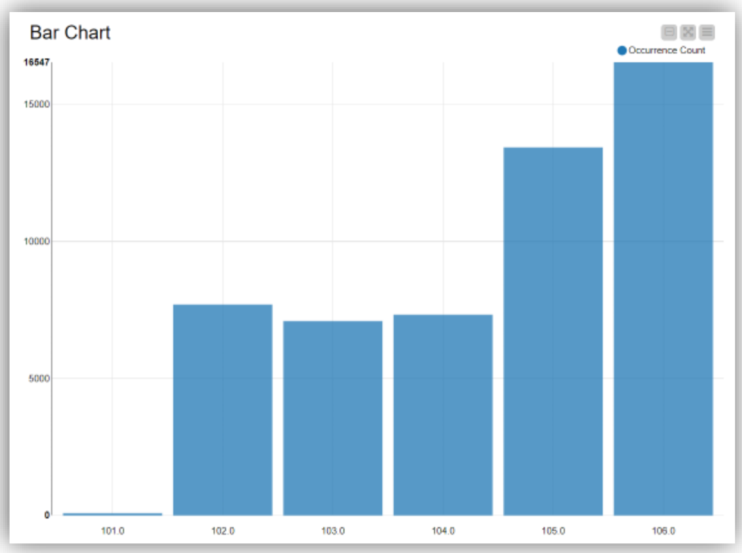


Figure 34. Bar chart for segment - row data

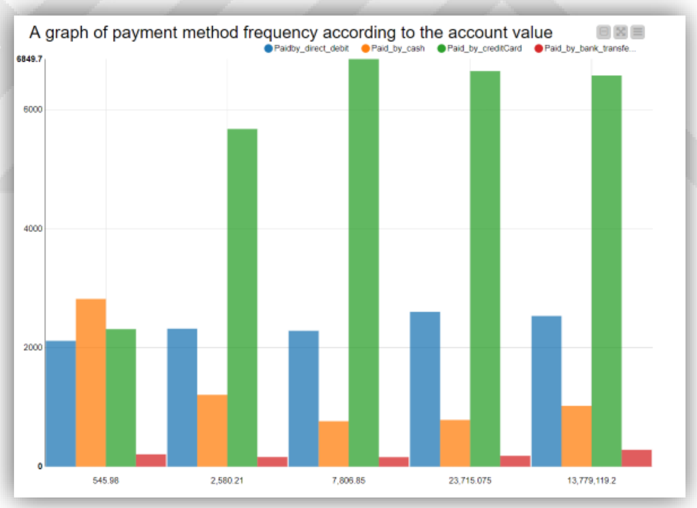


Figure 35. Payment method frequency according to account value(asset)

It was observed that the variable "average product age" is left-skewed, with a majority of customers having a yearly extra payment amount of zero. When the distribution of customers' assets according to their payment method was analyzed, it was found that credit cards were the dominant payment method for those with relatively high assets. However, when the "account_value_asset" was less than 545

TL, it was observed that this group preferred to make cash payments over other payment methods.

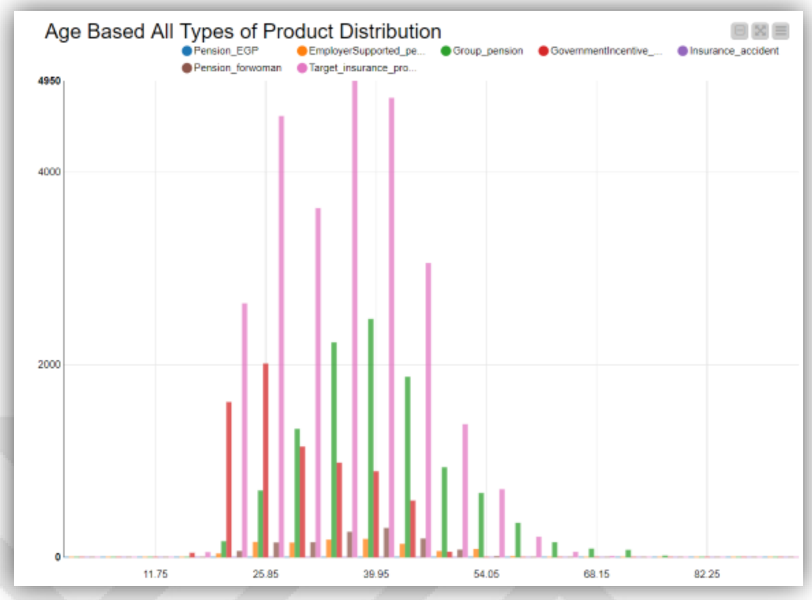


Figure 36. Number of various products concerning customer age

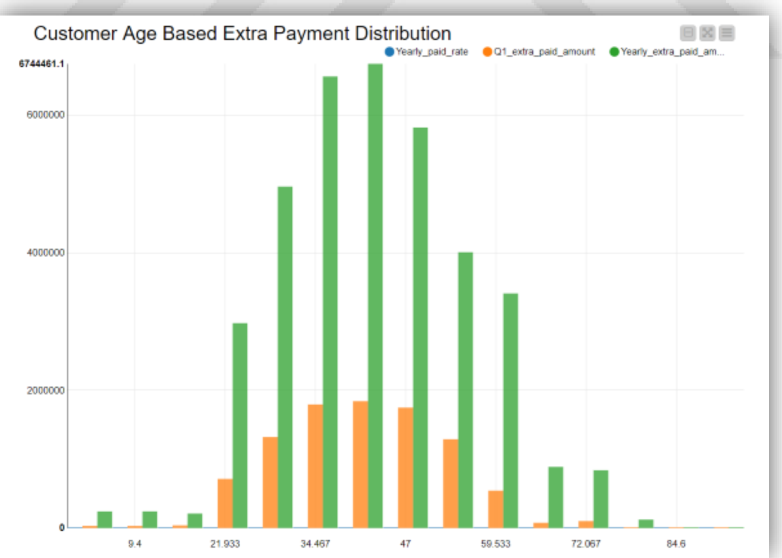


Figure 37. Product distribution and extra payment behavior concerning age

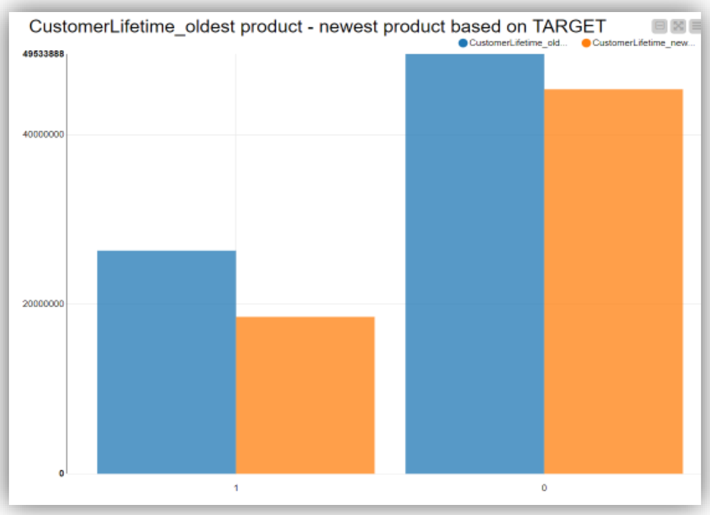


Figure 38. Customer lifetime according target variable class

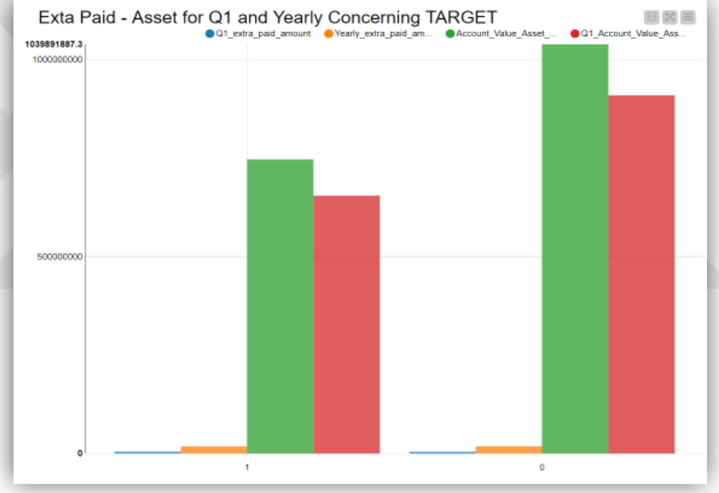


Figure 39. Asset and customer lifetime concerning TARGET class

When segmented according to age, it was found that the "elite" segment predominantly used credit cards for payment. When the distribution of customers with pension and insurance products was analyzed according to age, it was found that the highest ownership of these products was in the 25-35 age range. The "government incentive pension product" was also predominantly found in the younger group. This distribution was likely due to the fact that this product is mandatory for those who are just starting their careers. It was also found that customers who do not have the target product have been customers of the company for a longer period of time.

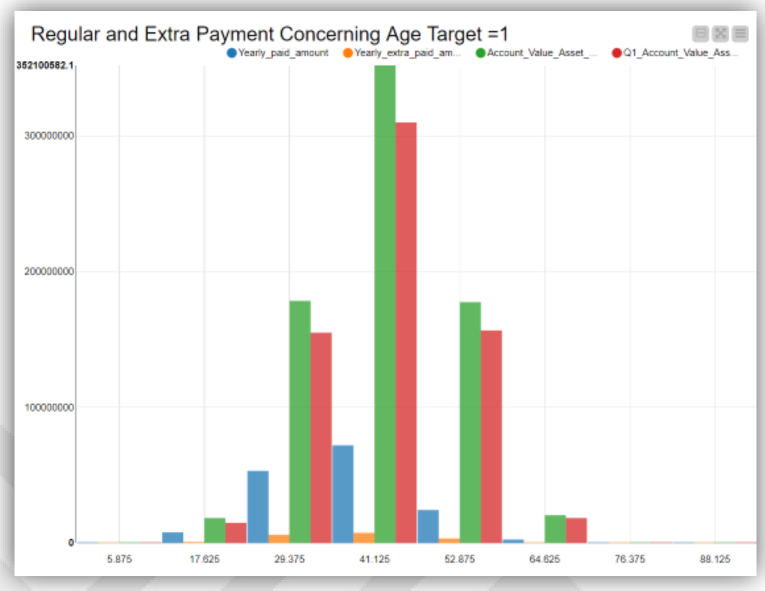


Figure 40. Payment behaviours: Extra and regular payment, account value concerning customer age when target variable = 1

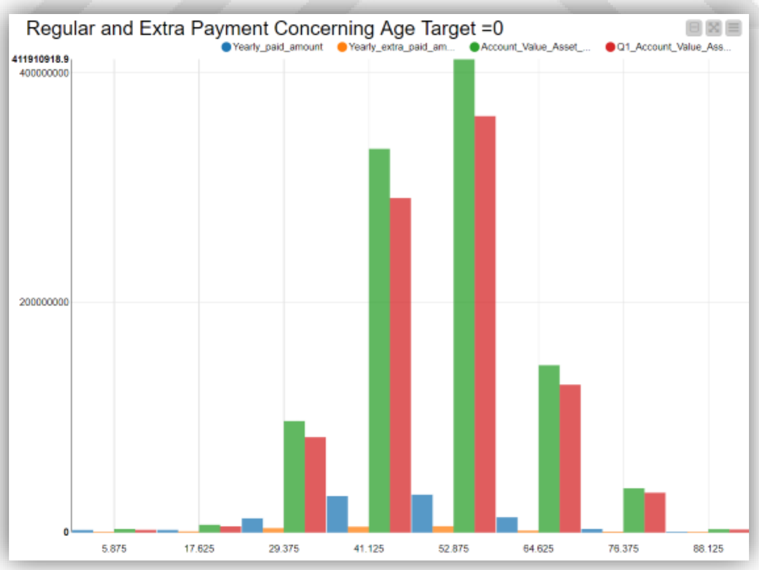
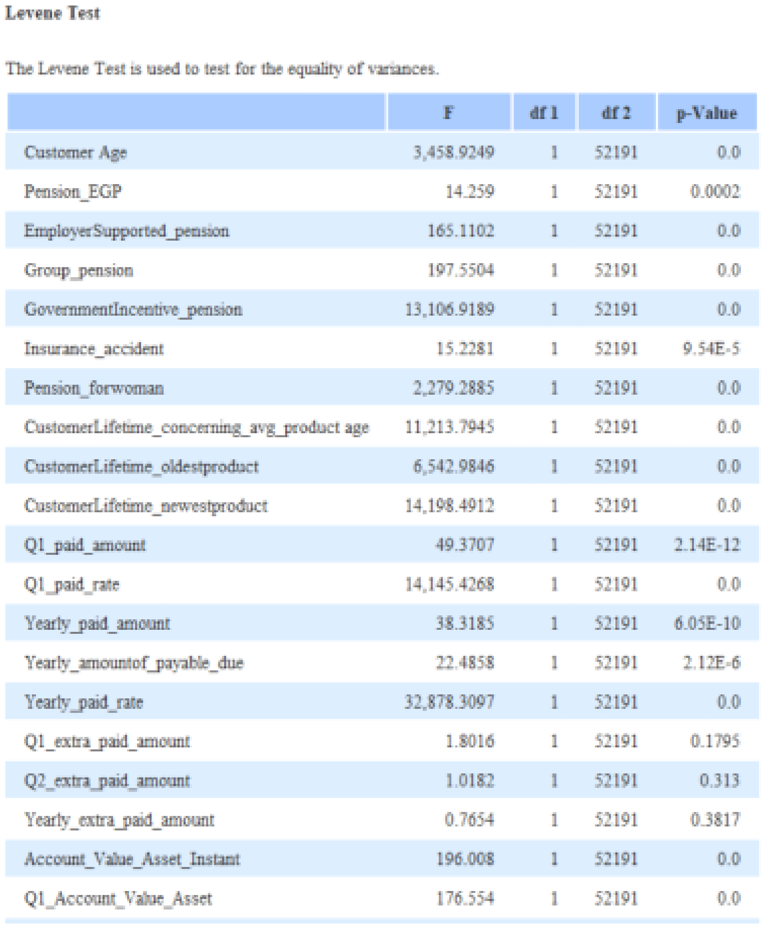


Figure 41. Payment behaviours: Extra and regular payment distribution concerning customer age when target variable = 0

The customer segment was determined to be left-skewed, with the elite customer segment comprising a minority of the customer portfolio. The low segment was identified as the densest group. The distribution of customer age was observed to be close to normal. An examination of customer payment behavior suggested that customers either effectively manage their payments or do not make payments at all, as indicated by the rate of paid.

After conducting a histogram analysis of the data set, it was observed that the distribution of customer product age displayed a right tail. In addition, the product ownership analysis indicated that individuals with different product groups typically either had no product or only one product. Furthermore, the number of customers with multiple copies of the same product was found to be low.

In this study, the ANOVA test was used to determine whether there is a significant difference between two groups of customers with and without a life insurance product, in terms of the continuous variables under investigation. The Levene statistic was used to test the assumption of equal variance between the two groups, and a confidence interval of 95% was chosen for the two-sample comparison test.



Levene Test

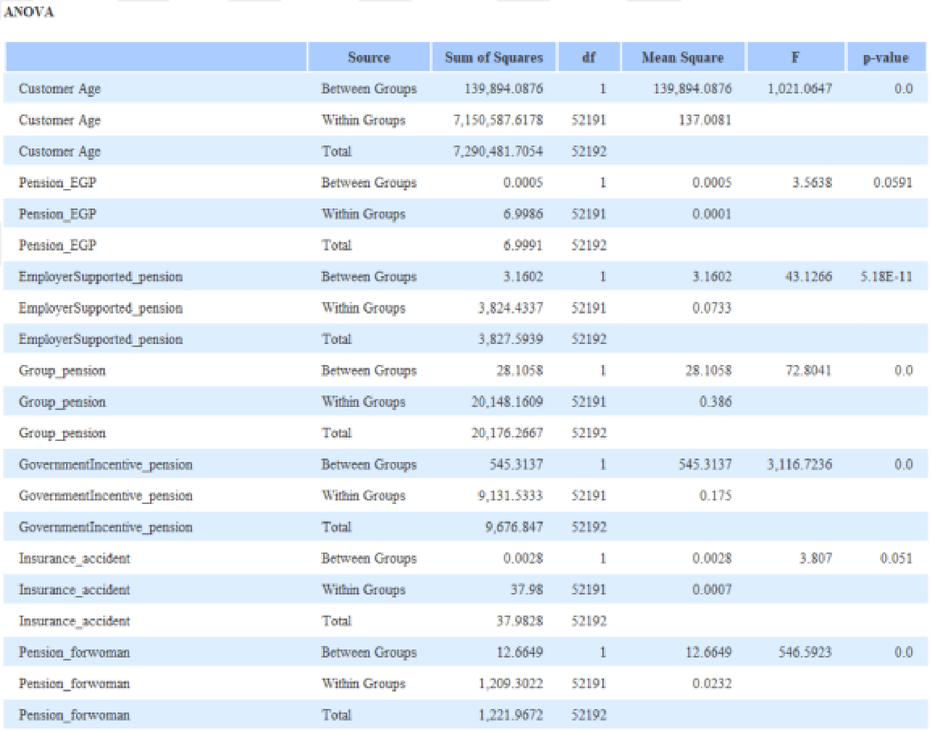
The Levene Test is used to test for the equality of variances.

	F	df 1	df 2	p-Value
Customer Age	3,458.9249	1	52191	0.0
Pension_EGP	14.259	1	52191	0.0002
EmployerSupported_pension	165.1102	1	52191	0.0
Group_pension	197.5504	1	52191	0.0
GovernmentIncentive_pension	13,106.9189	1	52191	0.0
Insurance_accident	15.2281	1	52191	9.54E-5
Pension_forwoman	2,279.2885	1	52191	0.0
CustomerLifetime_concerning_avg_product age	11,213.7945	1	52191	0.0
CustomerLifetime_oldestproduct	6,542.9846	1	52191	0.0
CustomerLifetime_newestproduct	14,198.4912	1	52191	0.0
Q1_paid_amount	49.3707	1	52191	2.14E-12
Q1_paid_rate	14,145.4268	1	52191	0.0
Yearly_paid_amount	38.3185	1	52191	6.05E-10
Yearly_amountof_payable_due	22.4858	1	52191	2.12E-6
Yearly_paid_rate	32,878.3097	1	52191	0.0
Q1_extra_paid_amount	1.8016	1	52191	0.1795
Q2_extra_paid_amount	1.0182	1	52191	0.313
Yearly_extra_paid_amount	0.7654	1	52191	0.3817
Account_Value_Asset_Instant	196.008	1	52191	0.0
Q1_Account_Value_Asset	176.554	1	52191	0.0

Figure 42. Levene Test

The results of the ANOVA test showed that, for the variable "extra_paid_amount," the p-value was above 0.05, indicating that there was no significant difference between the two groups. However, for all other variables, the p-values were below 0.05, indicating that there were significant differences between the two groups. The null hypothesis, which states that the sample means are equal to the overall mean, was rejected in these cases.

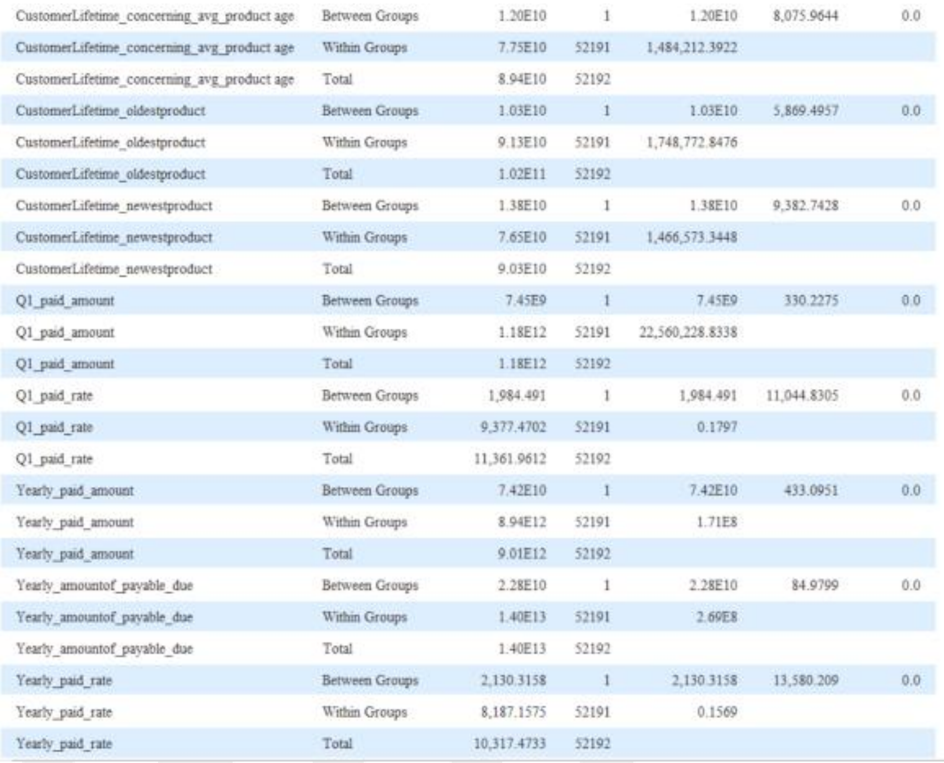
The F-value calculated in the ANOVA test was also found to be significant, indicating that at least one of the independent variables was important in differentiating between the two groups. Overall, these results suggest that there are significant differences between the two groups in terms of the variables being investigated, except for the variable "extra_paid_amount."



ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
Customer Age	Between Groups	139,894.0876	1	139,894.0876	1,021.0647	0.0
Customer Age	Within Groups	7,150,587.6178	52191	137.0081		
Customer Age	Total	7,290,481.7054	52192			
Pension_EGP	Between Groups	0.0005	1	0.0005	3.5638	0.0591
Pension_EGP	Within Groups	6.9986	52191	0.0001		
Pension_EGP	Total	6.9991	52192			
EmployerSupported_pension	Between Groups	3.1602	1	3.1602	43.1266	5.18E-11
EmployerSupported_pension	Within Groups	3,824.4337	52191	0.0733		
EmployerSupported_pension	Total	3,827.5939	52192			
Group_pension	Between Groups	28.1058	1	28.1058	72.8041	0.0
Group_pension	Within Groups	20,148.1609	52191	0.386		
Group_pension	Total	20,176.2667	52192			
GovernmentIncentive_pension	Between Groups	545.3137	1	545.3137	3,116.7236	0.0
GovernmentIncentive_pension	Within Groups	9,131.5333	52191	0.175		
GovernmentIncentive_pension	Total	9,676.847	52192			
Insurance_accident	Between Groups	0.0028	1	0.0028	3.807	0.051
Insurance_accident	Within Groups	37.98	52191	0.0007		
Insurance_accident	Total	37.9828	52192			
Pension_forwoman	Between Groups	12.6649	1	12.6649	546.5923	0.0
Pension_forwoman	Within Groups	1,209.3022	52191	0.0232		
Pension_forwoman	Total	1,221.9672	52192			

Figure 43. ANOVA-One Way Table



CustomerLifetime_concerning_avg_product age	Between Groups	1.20E10	1	1.20E10	8,075.9644	0.0
CustomerLifetime_concerning_avg_product age	Within Groups	7.75E10	52191	1,484,212.3922		
CustomerLifetime_concerning_avg_product age	Total	8.94E10	52192			
CustomerLifetime_oldestproduct	Between Groups	1.03E10	1	1.03E10	5,869.4957	0.0
CustomerLifetime_oldestproduct	Within Groups	9.13E10	52191	1,748,772.8476		
CustomerLifetime_oldestproduct	Total	1.02E11	52192			
CustomerLifetime_newestproduct	Between Groups	1.38E10	1	1.38E10	9,382.7428	0.0
CustomerLifetime_newestproduct	Within Groups	7.65E10	52191	1,466,573.3448		
CustomerLifetime_newestproduct	Total	9.03E10	52192			
Q1_paid_amount	Between Groups	7.45E9	1	7.45E9	330.2275	0.0
Q1_paid_amount	Within Groups	1.18E12	52191	22,560,228.8338		
Q1_paid_amount	Total	1.18E12	52192			
Q1_paid_rate	Between Groups	1,984.491	1	1,984.491	11,044.8305	0.0
Q1_paid_rate	Within Groups	9,377.4702	52191	0.1797		
Q1_paid_rate	Total	11,361.9612	52192			
Yearly_paid_amount	Between Groups	7.42E10	1	7.42E10	433.0951	0.0
Yearly_paid_amount	Within Groups	8.94E12	52191	1.71E8		
Yearly_paid_amount	Total	9.01E12	52192			
Yearly_amountof_payable_due	Between Groups	2.28E10	1	2.28E10	84.9799	0.0
Yearly_amountof_payable_due	Within Groups	1.40E13	52191	2.69E8		
Yearly_amountof_payable_due	Total	1.40E13	52192			
Yearly_paid_rate	Between Groups	2,130.3158	1	2,130.3158	13,580.209	0.0
Yearly_paid_rate	Within Groups	8,187.1575	52191	0.1569		
Yearly_paid_rate	Total	10,317.4733	52192			

Figure 44. Anova (cont.d)

4.8.1 Target variable age distribution

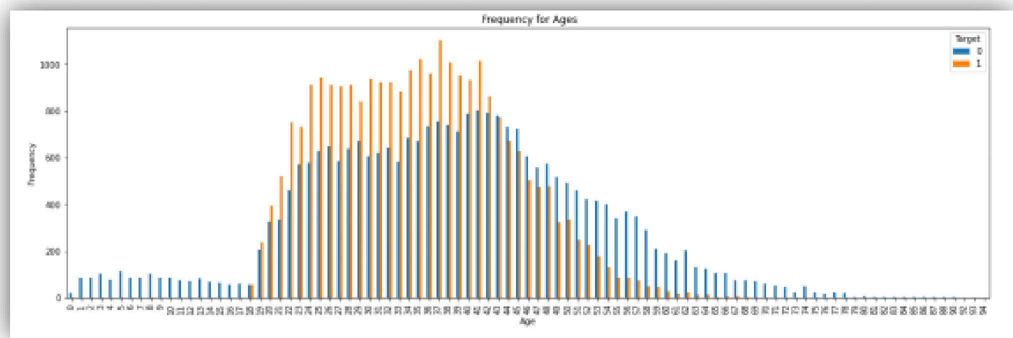


Figure 45. Frequency of target variable concerning customer age

A study examining the age distribution of customers with and without life insurance has revealed a decrease in the number of individuals with the product after the age of 46. A concentration of individuals in the age range of 21-43 was also observed. This trend was also seen among individuals without the product. One possible explanation for this phenomenon is that individuals with health issues may face restrictions when purchasing life insurance. Healthy individuals can more easily obtain life insurance, but as individuals age, the proportion of those with health problems may increase, leading to a decrease in the number of individuals with life insurance in older age groups.

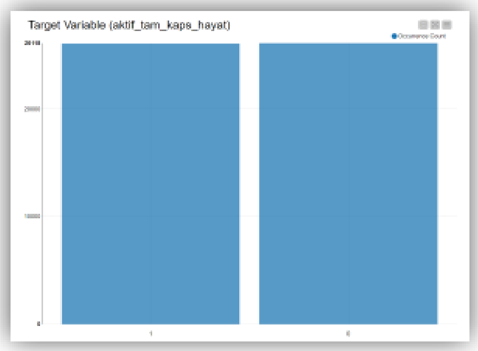


Figure 46. Distribution of target class

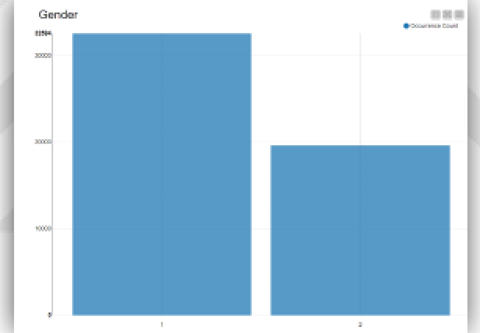


Figure 47.Distribution of gender

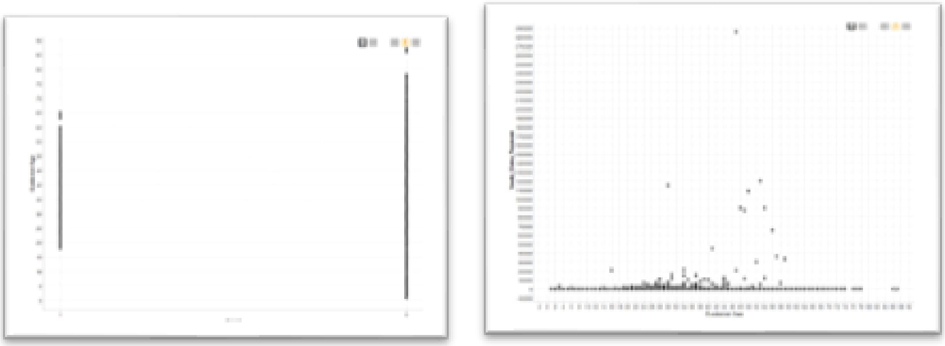


Figure 48. Scatter plot age-target, age-extra payment

Scatter plots were used for pair analysis. According to these graphs, customers who have the target product are in the age range of 20-65. Customers in the age range of 46-60 can make extra payments in addition to their regular payments. The customers with the highest amount of assets in their accounts are in the age range of 46-54.

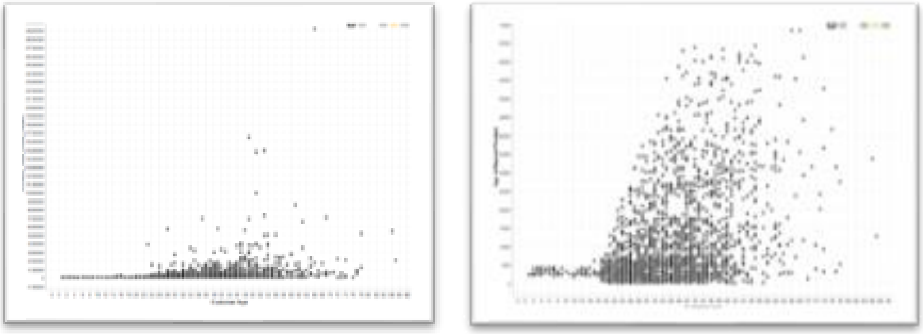


Figure 49. Scatter plot customer age-instant_asset, customer age- product age

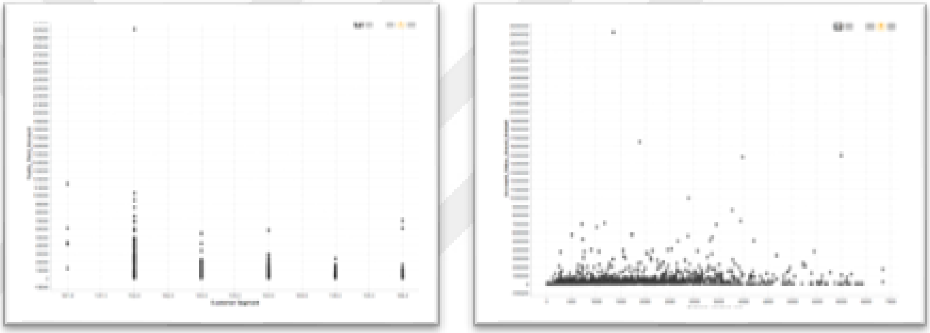


Figure 50. Scatter plot segment-payment, asset- avg. product age

The segment that makes the most payments is not the elite but the group which is called as a special group. A concentration of customers who have been in the system for less than 3,000 days was observed in the group with pension products.

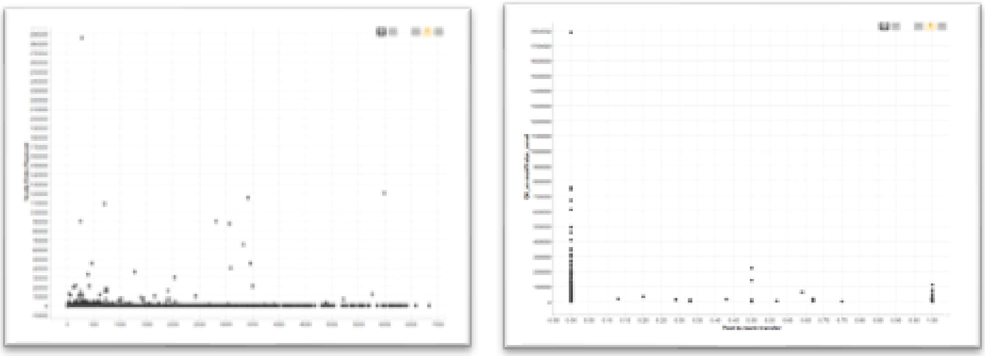


Figure 51. Extra_payment-avg.product_age, paid by bank transfer- Q4asset

It was observed that the group of customers who were found in the system for less than 1,000 days had made more extra payments. The group of customers who used a payment method other than cash had a higher total asset in their accounts.

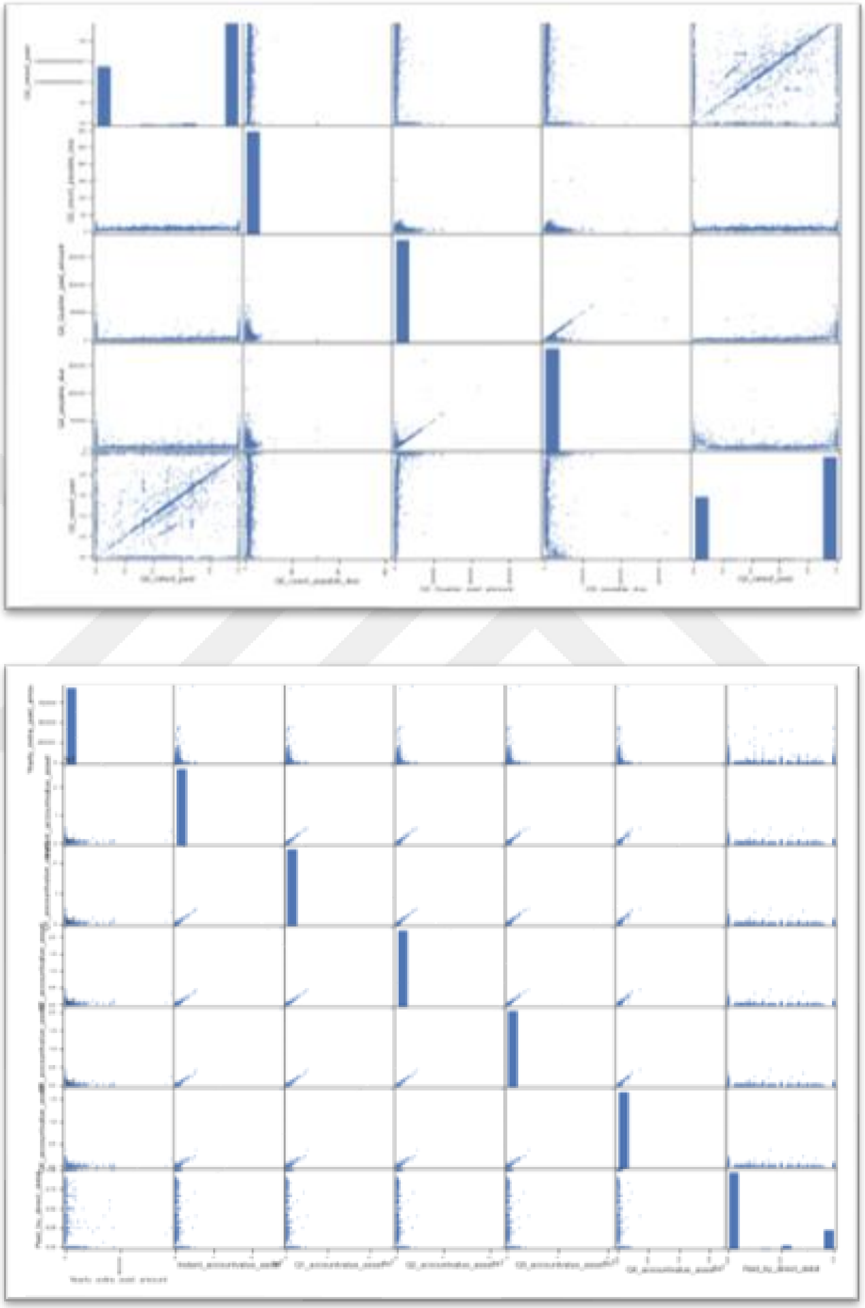


Figure 52. Scatter plot for multi variables

4.9 Correlation Matrix

Spearman's rank correlation was applied to all continuous data in the data set. The analysis revealed a high degree of correlation between customer payment behavior. Additionally, high correlations (around 90%) were also found between the customer product age and average age features. To reduce the dimensionality of the data, only one of the highly correlated features was selected and used in the modeling process, while all of the features were also tested in different model scenarios. The performance of the resulting models was compared to evaluate their effectiveness.

Table 5

Spearman's Rho Rank Correlation

Variable 1	Variable 2	Correlation
Average_age_of_productlife_forAll	Age_of_oldest_product	1.00
Average_age_of_productlife_forAll	Age_of_newest_product	1.00
Age_of_oldest_product	Age_of_newest_product	1.00
Q1_rateof_paid	yearly_rateof_paid	0.99
Instant_accountvalue_asset	Q1_accountvalue_asset	0.97
Q4_count_payable_due	yearly_count_payable	0.96
Q3_accountvalue_asset	Q4_accountvalue_asset	0.95
Q2_accountvalue_asset	Q3_accountvalue_asset	0.94
Q3_payable_due	yearly_payable_due	0.94

Table 5 (cont.d)

Variable 1	Variable 2	Correlation
Q3_Quarter_paid_amount	yearly_paid_amount	0.92
Q1_accountvalue_asset	Q2_accountvalue_asset	0.93
Q1_payableamount_due	Q2_payable_due	0.92
Instant_accountvalue_asset	Q2_accountvalue_asset	0.90
Q2_Quarter_paid_amount	yearly_paid_amount	0.90
Q2_accountvalue_asset	Q4_accountvalue_asset	0.89
Q2_payable_due	yearly_payable_due	0.89
Q4_payable_due	yearly_payable_due	0.88
Q1_accountvalue_asset	Q3_accountvalue_asset	0.87
Q1_paid_amount	Q2_Quarter_paid_amount	0.86
Q2_payable_due	Q3_payable_due	0.85
Q2_Quarter_paid_amount	Q3_Quarter_paid_amount	0.85
Q4_paid_amount	yearly_paid_amount	0.85
Q1_paid_amount	yearly_paid_amount	0.85
Instant_accountvalue_asset	Q3_accountvalue_asset	0.84

Table 5 (cont.d)

Variable 1	Variable 2	Correlation
Q1_payableamount_due	yearly_payable_due	0.83
Q3_Quarter_paid_amount	Q4_paid_amount	0.82
Q3_payable_due	Q4_payable_due	0.82
Q4_paid_amount	Q4_payable_due	0.82
Q1_accountvalue_asset	Q4_accountvalue_asset	0.82
Q2_rateof_paid	yearly_rateof_paid	0.81
Q3_Quarter_paid_amount	Q3_payable_due	0.80
Q1_rateof_paid	Q2_rateof_paid	0.79
Instant_accountvalue_asset	Q4_accountvalue_asset	0.79
Q1_payableamount_due	Q3_payable_due	0.78
Q2_Quarter_paid_amount	Q2_payable_due	0.78
yearly_paid_amount	yearly_payable_due	0.75
Q3_Quarter_paid_amount	yearly_payable_due	0.74
Q2_rateof_paid	Q3_rateof_paid	0.74

Table 5 (cont.d)

Variable 1	Variable 2	Correlation
Q4_paid_amount	Q4_rateof_paid	0.74
Q1_paid_amount	Q3_Quarter_paid_amount	0.73
Q4_paid_amount	yearly_payable_due	0.72
Q1_payableamount_due	Q2_Quarter_paid_amount	0.72
Q3_count_payable_due	yearly_count_payable	0.72
Q3_rateof_paid	Q4_rateof_paid	0.70
Q1_paid_amount	Q1_payableamount_due	0.70
Q3_payable_due	yearly_paid_amount	0.70
Q4_payable_due	Q4_count_payable_due	0.70
Age_of_oldest_product	Q4_accountvalue_asset	0.70
Average_age_of_productlife_forAll	Q4_accountvalue_asset	0.70
Age_of_newest_product	Q4_accountvalue_asset	0.69
Average_age_of_productlife_forAll	Q1_accountvalue_asset	0.68
Age_of_oldest_product	Q1_accountvalue_asset	0.68

Table 5 (cont.d)

Variable 1	Variable 2	Correlation
Age_of_newest_product	Q1_accountvalue_asset	0.68
Average_age_of_productlife_forAll	Instant_accountvalue_asset	0.68
Age_of_oldest_product	Instant_accountvalue_asset	0.68
Age_of_newest_product	Instant_accountvalue_asset	0.68
Average_age_of_productlife_forAll	Q3_accountvalue_asset	0.68
Age_of_oldest_product	Q3_accountvalue_asset	0.68
Q2_payable_due	Q3_Quarter_paid_amount	0.68
Q1_paid_amount	Q1_rateof_paid	0.68
Age_of_newest_product	Q3_accountvalue_asset	0.68
Q2_payable_due	yearly_paid_amount	0.67
Q2_Quarter_paid_amount	Q4_paid_amount	0.67

4.10 Decision Tree Application

The decision tree model has been implemented on the insurance dataset with a target variable 0-1. Quality measure selected as Gini index introduced by (Gini, 1912):

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

In decision trees, the Gini index was used to evaluate splits in data and determine which split produces a greater separation between classes. The Gini index enables us to observe how well a given model divides data into classes and thus assess its accuracy. A low Gini index indicates that most of the cases were concentrated in one class, on the other hand, a high Gini index implies more diversity across classes (Rokach & Maimon, 2006). The number of threats taken is 12 and the minimum number of records per node applied was 2. In the first experiment with decision three, no pruning method made used on the dataset. In this experiment, the partition method of 30% test set and 70% training set was taken. Hyperparameter tuning was limited to a range of 2-15 nodes.

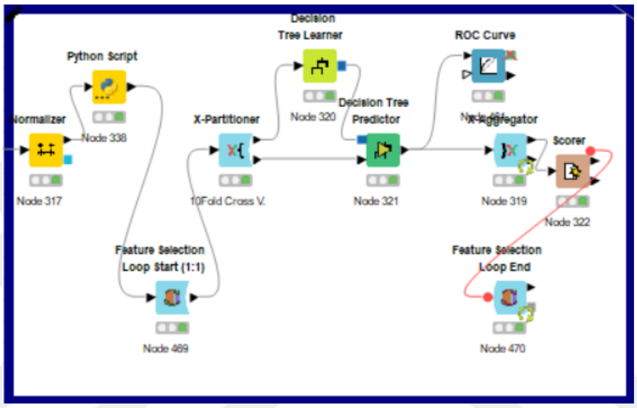


Figure 53. A decision tree structure with K-fold

In the initial trial, no normalization process was applied to the data. As data pre-processing, outliers were removed, and the target variable was taken as balanced data for modeling. Furthermore, the dimension reduction technique has not been employed and all dataset is included in the model. In the second model applied, the data was split into two parts, train, and test, with a ratio of 30% to 70%, respectively. All variables were kept constant from the first model except for the pruning method which was set to Minimum Description Length (MDL). The general model architecture for decision trees was structured as shown in Figure 36. In the second model, the test-train split model was set to 30% test and 70% train set, and the maximum depth of the tree was selected as 15 using Knime Parameter Optimization Loop.

Table 6

Applied Methods on Decision Tree

Algorithm	Model	Normalization	10Fold Cross Validation	30/70 Test/Train	Feature Selection	Pruning	Hyperparameter Tuning
Decision Tree	M1	-	-	+	-	-	+
	M2	-	-	+	-	+	+
	M3	+	-	+	-	+	+
	M4	-	+	-	-	+	-
	M5	-	+	-	-	-	-
	M6	+	+	-	+	+	-
	M7	+	-	+	-	-	-

In the third model architecture, the effect of normalization on the dataset was measured on the model performance. To this end, the dataset from the second model architecture was divided into test trains and pruning and hyperparameter optimization were kept at a maximum depth of tree 15, and the normalization process was applied to variables. In the fourth decision tree modeling, the effect of K-fold cross-validation on model performance was measured. For this purpose, this time the dataset partitioning has not been divided into 30% test and 70% train. 10-fold cross-validation was applied. For the first application, pruning The Minimum Description Length (MDL) method was not carried out, in the second one it was added to the architecture as a pruning method. In the 5th model 10Fold, cross-validation was applied as illustrated in Table. Model 6 represents all possible optimization methods for the model. The forward feature selection method was implemented and the 10Fold cross-validation method was used. Through the Model 7 structure, the effect of normalization on the model has been observed while keeping all other values constant.

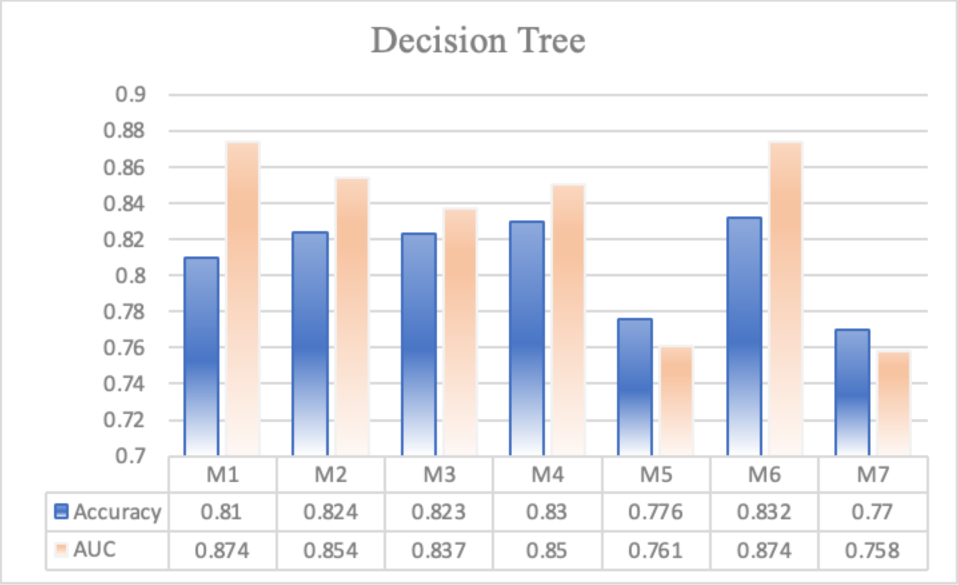


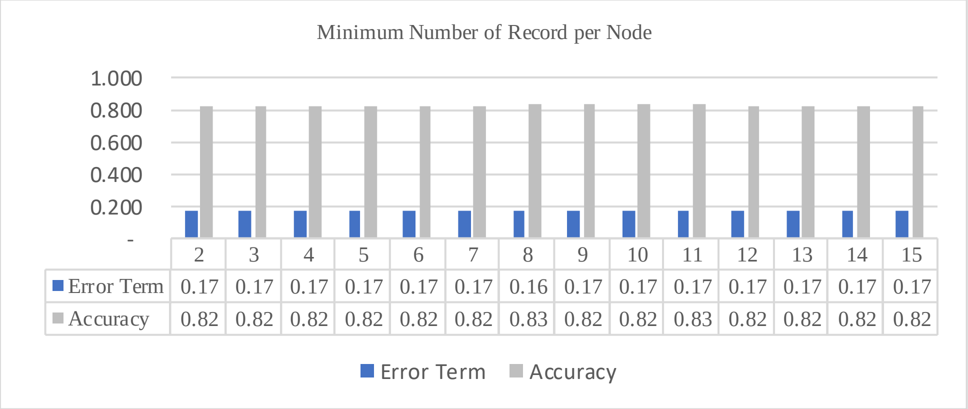
Figure 54. Decision tree experiment setup AUC

Table 7

Model performance scores decision tree

Model	Class	TP	FP	TN	FN	Recall	Precision	Sensitivity	Specificity	F-Score	Accuracy	Cohen's kappa
M1	0	519	139	1208	267	0.66	0.79	0.66	0.9	0.72		
	1	1208	267	519	139	0.9	0.82	0.9	0.66	0.86		
	Overall										0.81	0.57
M2	0	559	148	1199	227	0.71	0.79	0.71	0.89	0.75		
	1	1199	227	559	148	0.89	0.84	0.89	0.71	0.86		
	Overall										0.824	0.61
M3	0	560	151	1196	226	0.71	0.79	0.71	0.89	0.75		
	1	1196	226	560	151	0.89	0.84	0.89	0.71	0.86		
	Overall										0.823	0.61
M4	0	1887	473	4015	732	0.72	0.8	0.72	0.89	0.76		
	1	4015	732	1887	473	0.89	0.85	0.89	0.72	0.87		
	Overall										0.83	0.63
M5	0	1825	799	3689	794	0.7	0.7	0.7	0.82	0.7		
	1	3689	794	1825	799	0.82	0.82	0.82	0.7	0.82		
	Overall										0.776	0.52
M6	0	1870	451	4037	749	0.71	0.81	0.71	0.9	0.76		
	1	4037	749	1870	451	0.9	0.84	0.9	0.71	0.87		
	Overall										0.831	0.63
M7	0	532	231	1120	250	0.68	0.7	0.68	0.83	0.69		
	1	1120	250	532	231	0.83	0.82	0.83	0.68	0.82		
	Overall										0.77	0.51

In the second model structure, the pruning method was selected as Minimum Description Length, unlike the first model. It was observed that this change resulted in a 0.01-point increase in the accuracy score while causing a decrease in the AUC score. In both models, the minimum number of records per node was selected between 2-5 for parameter optimization.



	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Error Term	0.17	0.17	0.17	0.17	0.17	0.17	0.16	0.17	0.17	0.17	0.17	0.17	0.17	0.17
Accuracy	0.82	0.82	0.82	0.82	0.82	0.82	0.83	0.82	0.82	0.83	0.82	0.82	0.82	0.82

Figure 55. Hyperparameter optimization

To minimize the error term, it was found that the best results were obtained when the minimum number of records was 8. The accuracy score also gave the most optimal result for the value of 8. In the third model, it was determined that the normalization process applied to the data set had no positive effect on the model performance. The effect of 10fold cross-validation on the model was evaluated in the 4th model. In this way, the model can use more data as train data. It was tested that it increased compared to the 3rd model. However, it did not cause a significant improvement in model performance.

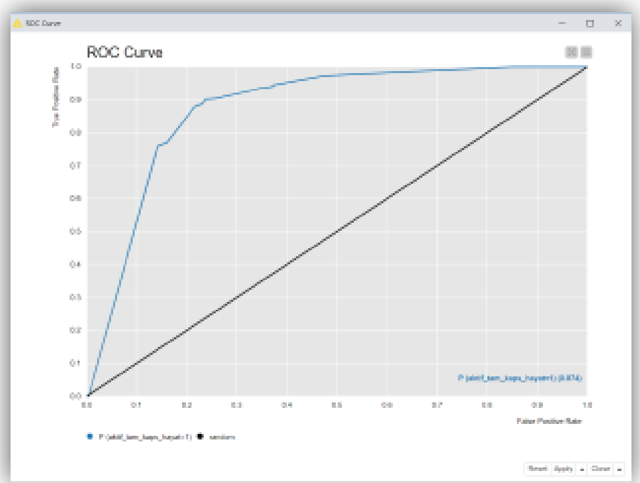


Figure 56. ROC curve for model6

In the 5th model, pruning, normalization, and hyperparameter optimization were not performed, only 10-fold cross-validation was applied. In this case, a 0.1-point drop was observed in model performance, which can be considered significant compared to other methods. In the 6th model, pruning and forward feature selection were applied, as well as 10-fold cross-validation, resulting in an improvement in model performance. As a result of all these experiments, it was determined that the pruning process significantly contributed to the model performance.

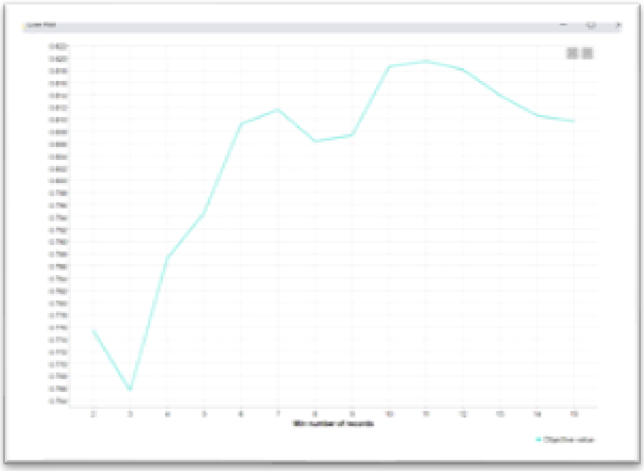


Figure 57. Accuracy score concerning depth of tree

Table 8

Forward Feature Selection for Decision Tree

Number of Features	Added Feature	Accuracy
1	Q2_Quarter_paid_amount	0.77
2	Q3_Quarter_paid_amount	0.78
3	Q1_Extra_payment	0.78
4	Customer_segment	0.78
5	Q3_accountvalue_asset	0.80
6	PensionproductOwner_iges	0.81
7	Paid_by_cash	0.81
8	Pension_std_group	0.80
9	Pension_katilim_group	0.81
10	Instant_accountvalue_asset	0.81
11	Productowner_SKH_ insurance	0.81
12	Productowner_OKS_pension	0.81
13	Productowner_forwoman_pension	0.81
14	Q3_count_payable_due	0.82
15	Q2_extra_paid_amount	0.81
16	ProductOwner_group_pension	0.81
17	ProductOwner_forchildren_pension	0.81
18	Q1_accountvalue_asset	0.81
19	ProductOwner_vasa_egp_pension	0.81
20	ProductOwner_personal_accident_insurance	0.81

Forward feature selection is a method of dimension reduction used in decision tree modeling to evaluate the impact of variables on the model's explanatory power. In this method, even though the number of variables in the model increases, the accuracy score remains within the range of 0.77-0.82. It has been observed that the variable Q3_accountvalue_asset has an impact on the accuracy score, as illustrated in Table 8.

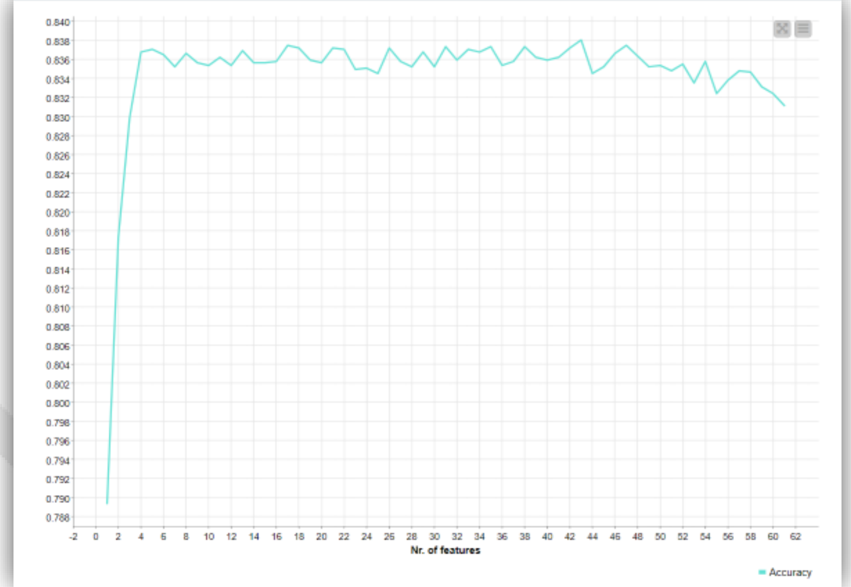


Figure 58. Forward feature selection

4.11 Logistic Regression Application

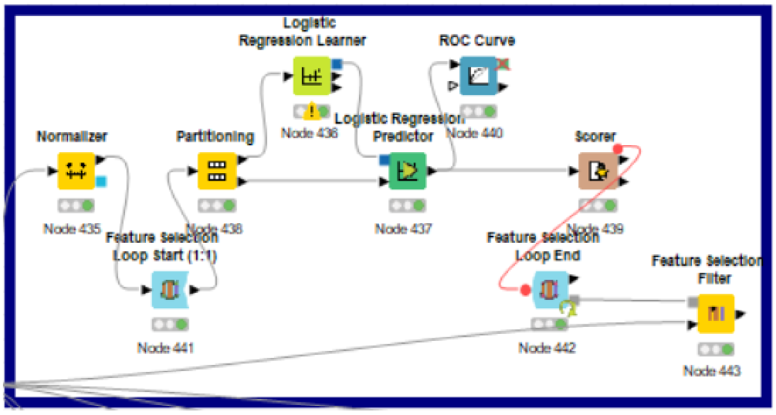


Figure 59. Logistic regression model structure

To solve a classification problem using logistic regression, it is necessary to check whether the logistic regression assumptions are satisfied. One of these assumptions is that the dataset should not contain missing values, as logistic regression cannot handle such data. In this study, the effect of removing missing values on the performance of the model was tested using 8 different models. In addition, the impact of normalization on logistic regression-based classification modeling was evaluated. To avoid multicollinearity among the independent variables, which is an important assumption in logistic regression, dimension reduction using Spearman's rank correlation was applied to features with a correlation greater than 0.6. The contribution of increasing the Epoch value in the hyperparameter optimization step to the model performance was also measured. The performance of the 8 models was compared based on their accuracy and AUC values.

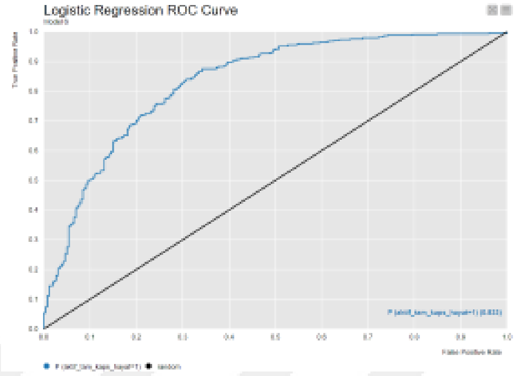
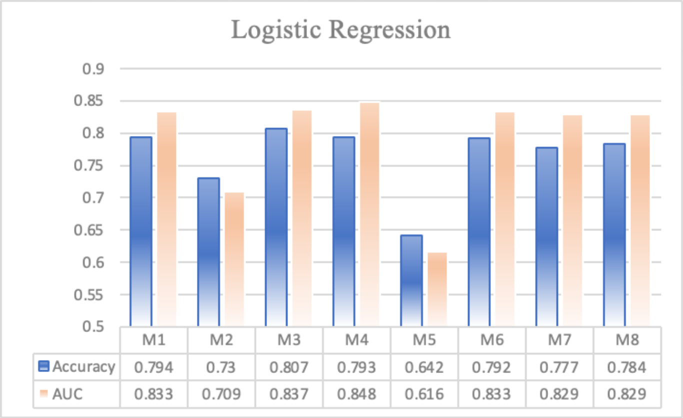


Figure 60. Logistic regression ROC



	M1	M2	M3	M4	M5	M6	M7	M8
▇ Accuracy	0.794	0.73	0.807	0.793	0.642	0.792	0.777	0.784
▇ AUC	0.833	0.709	0.837	0.848	0.616	0.833	0.829	0.829

Figure 61. Logistic regression AUC

In the first developed model, the normalized data set was used as the baseline, and the data was split into a 70% training set and a 30% test set. Forward feature elimination was applied, and dimension reduction was not performed on the correlated data. The epoch value was set to 100 for hyperparameter tuning. In the second model, the data was not normalized, unlike the first model. It was observed that both the AUC and accuracy scores of the model with normalization were better than those of the model without normalization. In the third model, the epoch value was increased from 100 to 200. As a result, no change in accuracy was observed, but an increase in AUC was detected. In the fourth model, the data split method was changed. Instead of splitting the data into a 70% training set and a 30% test set, 10-fold cross-validation was applied, and the data was normalized. While no change in accuracy was observed, the AUC score increased.

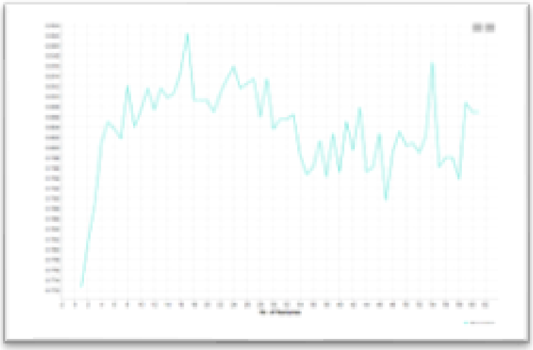


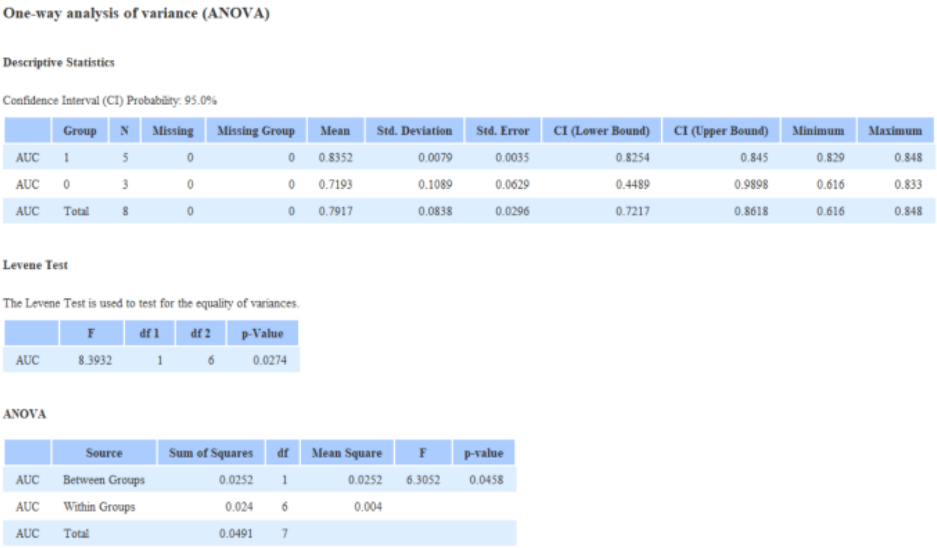
Figure 62. Forward feature selection

Table 9

Applied Methods on Logistic Regression

Algorithm	Model	Normalization	10FoldCross Validation	30/70 Test/Train	Forward Feature Selection	Epoch	Dimension Reduction via Correlation
Logistic Regression	M1	+	-	+	+	100	-
	M2	-	-	+	+	100	-
	M3	+	-	+	+	200	-
	M4	+	+	-	-	100	-
	M5	-	+	-	-	100	-
	M6	-	+	-	-	200	-
	M7	+	-	+	-	100	+
	M8	+	+	-	-	100	+

Thus, it was observed that increasing the epoch, normalization, and 10-fold cross-validation had a positive effect on model performance. In the fifth model, the epoch was set to 100 without normalization, and a significant decrease in both the model accuracy and AUC values was observed. In the sixth model, the epoch value was increased from 100 to 200. As a result, a significant increase in both the AUC and model accuracy scores was obtained, even without normalization. In models 7 and 8, it was observed that normalization did not significantly affect the model performance, regardless of the data split method.



One-way analysis of variance (ANOVA)

Descriptive Statistics

Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation	Std. Error	CI (Lower Bound)	CI (Upper Bound)	Minimum	Maximum
AUC	1	5	0	0	0.8352	0.0079	0.0035	0.8254	0.845	0.829	0.848
AUC	0	3	0	0	0.7193	0.1089	0.0629	0.4489	0.9898	0.616	0.833
AUC	Total	8	0	0	0.7917	0.0838	0.0296	0.7217	0.8618	0.616	0.848

Levene Test

The Levene Test is used to test for the equality of variances.

	F	df 1	df 2	p-Value
AUC	8.3932	1	6	0.0274

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
AUC	Between Groups	0.0252	1	0.0252	6.3052	0.0458
AUC	Within Groups	0.024	6	0.004		
AUC	Total	0.0491	7			

Figure 63. Anova table for logistic regression model normalization-AUC relation

Table 10

Logistic Regression Model Performance

Model	Class	TP	FP	TN	FN	Recall	Precision	Sensitivity	Specificity	F-Score	Accuracy	Cohen's kappa
M1	0	485	139	1208	301	0.62	0.78	0.62	0.90	0.69		
	1	1208	301	485	139	0.90	0.80	0.90	0.62	0.85		
	Overall										0.794	0.54
M2	0	493	282	1065	293	0.63	0.64	0.63	0.79	0.63		
	1	1065	293	493	282	0.79	0.78	0.79	0.63	0.79		
	Overall										0.730	0.42
M3	0	506	132	1215	280	0.64	0.79	0.64	0.90	0.71		
	1	1215	280	506	132	0.90	0.81	0.90	0.64	0.86		
	Overall										0.807	0.57
M4	0	1606	456	4032	1013	0.61	0.78	0.61	0.90	0.69		
	1	4032	1013	1606	456	0.90	0.80	0.90	0.61	0.85		
	Overall										0.793	0.54
M5	0	1441	1368	3120	1178	0.55	0.51	0.55	0.70	0.53		
	1	3120	1178	1441	1368	0.70	0.73	0.70	0.55	0.71		
	Overall										0.642	0.24
M6	0	1600	457	4031	1019	0.61	0.78	0.61	0.90	0.68		
	1	4031	1019	1600	457	0.90	0.80	0.90	0.61	0.85		
	Overall										0.792	0.53
M7	0	485	174	1173	301	0.62	0.74	0.62	0.87	0.67		
	1	1173	301	485	174	0.87	0.80	0.87	0.62	0.83		
	Overall										0.777	0.50
M8	0	1551	470	4018	1068	0.59	0.77	0.59	0.90	0.67		
	1	4018	1068	1551	470	0.90	0.79	0.90	0.59	0.84		
	Overall										0.784	0.51

In the forward feature selection process, it was determined that the 2ndQuarter_paid_amount, 3rdQuarter_paid_amount, Q1_extra_paid_amount, customer_segment, Q3_asset_portfolio, and group_product_bycompany_totheiremployee variables had an 80% effect on the model's success, and they significantly explained the model's prediction process.

4.12 XGBoosting Application

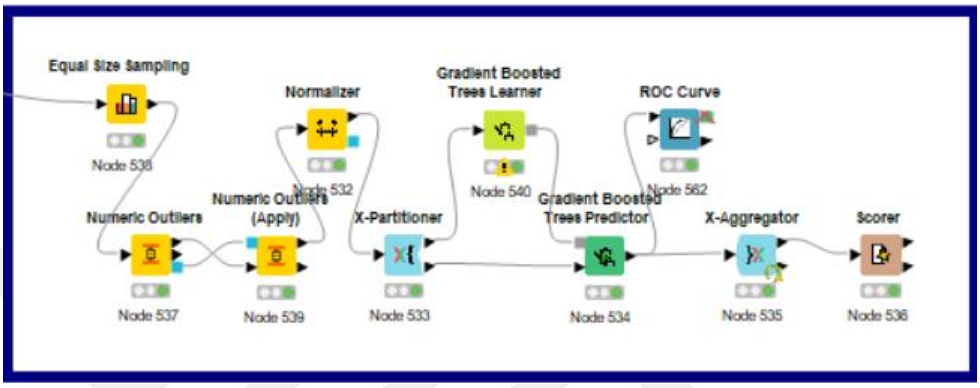


Figure 64. XGBoosting model structure

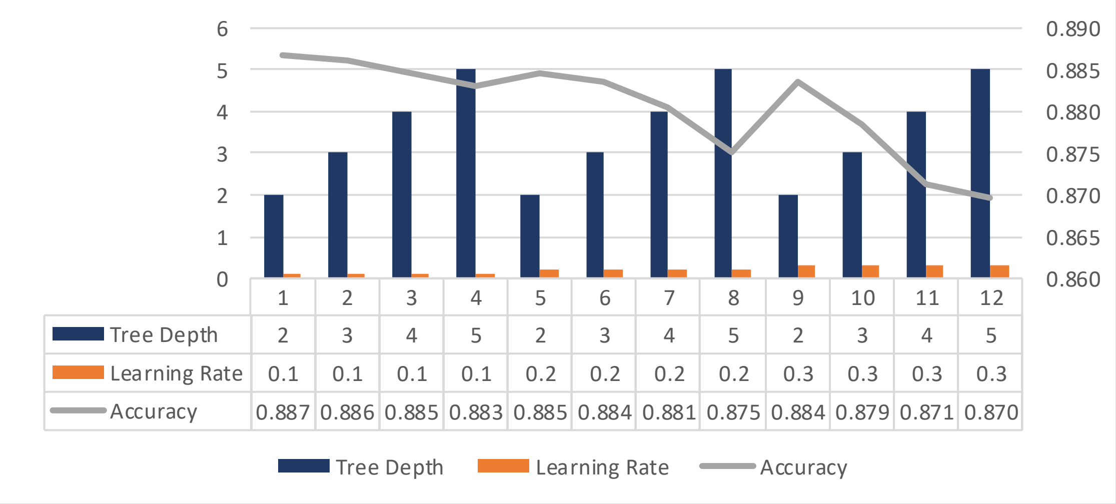
In a classification problem solved using the Gradient Boosted Tree algorithm, the effects of various data pre-processing steps and hyperparameter optimization on model performance were tested. The effect of normalization on the AUC and Accuracy score, as well as values such as precision, recall, and F1 score, were also examined for non-normally distributed data. The effects of splitting the data into test and train sets, with 30% as test and 70% as train, and using K-fold cross-validation were compared in terms of performance. Before building the model, the effects of handling missing values before modeling and allowing XGBoosting to handle these values without pre-processing were studied on model performance. Considering the model performance and computation time, backward feature elimination was applied for dimension reduction. It was found that even two variables may contain explanatory information as much as the other 62 variables for prediction. The effects of selecting the sampling method from among the same set of attributes and different sets of attributes were compared in terms of model performance. The effects of the maximum depth on model performance were also compared in terms of accuracy between tree depths of 1 and 5. In addition, the effects of changes in the learning rate hyperparameter on model performance were observed. When computing time was considered, it was observed that 10-fold cross-validation and backward feature elimination increased the processing time of the model by 20 times compared to the baseline model.

Table 11

Applied Methods on Gradient Boosted Tree

Algorithm	Model	Normalization	10Fold Cross Validation	30/70 Test/Train	Missing Value Omitted Before	Outlier Handling	Hyperparameter opt.
Gradient Boosted Trees	M1	-	-	+	+	+	The same set of attributes, max depth:4 boosting learning rate: 0.1
	M2	-	-	+	+	+	Attribute sampling=square root max depth: 4
	M3	-	+	-	+	+	The same set of attributes max dept: 4
	M4	-	+	-	+	+	A different set of attributes
	M5	-	+	-	+	+	Bagging option: fraction of data to learn single model = 1 with replacement
	M6	-	+	-	XGBoost	-	The same set of attributes, max depth:4
	M7	-	+	-	XGBoost	+	The same set of attributes, max depth:4
	M8	-	+	-	XGBoost	+	A different set of attributes max depth: 4
	M9	+	+	-	XGBoost	+	The same set of attributes, max depth:4
	M10	-	-	+	+	+	The same set of attributes, max depth:4, Boosting learning rate=0.3

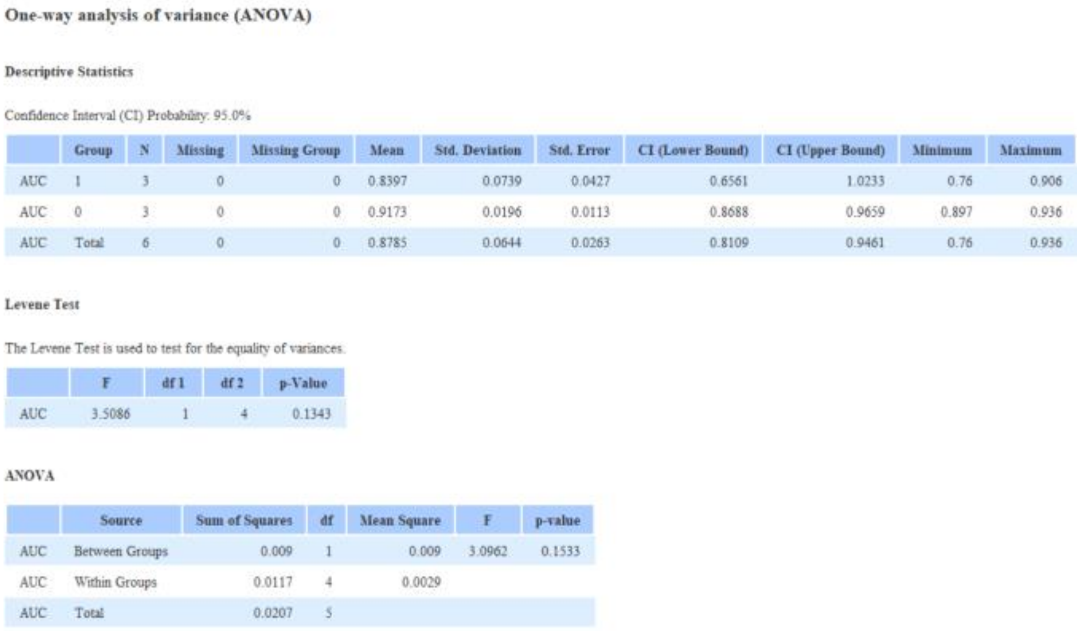
In this study of 10 different applications observed, the first model was divided into test and train sets as 30% and 70%, respectively. The data set was cleared of outliers and missing values. The maximum tree depth was set to 4, and the attribute selection was set to the same set of attributes with a learning rate of 0.1. In the second model, the split method was applied as K-fold cross-validation, and a decrease was observed in the AUC and accuracy scores. In the fourth model, 10-fold cross-validation was used again, but this time a different set of attribute sampling hyperparameter tuning methods was applied. In this case, the model performance was similar to that of the first and second models. In the fifth model, the Bagging option: fraction of data to learn single model = 1 with replacement was selected as the hyperparameter tuning method. This resulted in a decrease of approximately 0.2 points in the AUC. In the sixth and seventh models, no pre-processing was done for missing values and the gradient-boosted tree algorithm XGBoost was selected as the missing value handling method. A significant improvement in model performance was detected in this case



	1	2	3	4	5	6	7	8	9	10	11	12
Tree Depth	2	3	4	5	2	3	4	5	2	3	4	5
Learning Rate	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
Accuracy	0.887	0.886	0.885	0.883	0.885	0.884	0.881	0.875	0.884	0.879	0.871	0.870

Figure 65. XGBoosting tree depth and learning rate optimization

In the Gradient Boosted Tree algorithm, an increase in tree depth up to 5 was observed to increase the model's accuracy score during hyperparameter tuning. On the other hand, increasing the learning rate was observed to decrease the model's accuracy score.



One-way analysis of variance (ANOVA)

Descriptive Statistics

Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation	Std. Error	CI (Lower Bound)	CI (Upper Bound)	Minimum	Maximum
AUC	1	3	0	0	0.8397	0.0739	0.0427	0.6561	1.0233	0.76	0.906
AUC	0	3	0	0	0.9173	0.0196	0.0113	0.8688	0.9659	0.897	0.936
AUC	Total	6	0	0	0.8785	0.0644	0.0263	0.8109	0.9461	0.76	0.936

Levene Test

The Levene Test is used to test for the equality of variances.

	F	df 1	df 2	p-Value
AUC	3.5086	1	4	0.1343

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
AUC	Between Groups	0.009	1	0.009	3.0962	0.1533
AUC	Within Groups	0.0117	4	0.0029		
AUC	Total	0.0207	5			

Figure 66. Anova table for gradient boosting: missing value omitted -AUC relation

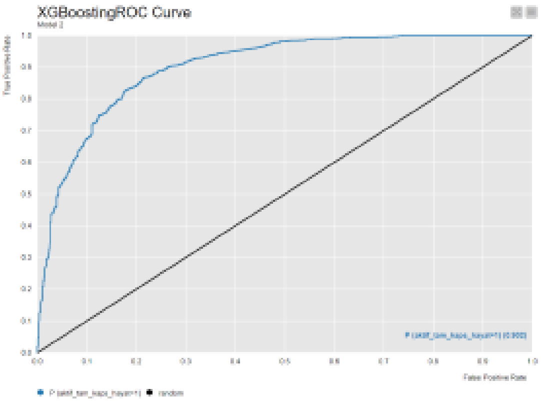


Figure 67. XGBoosting ROC curve

Table 12

Gradient Boosted Tree Model Performance

Model	Class	TP	FP	TN	FN	Recall	Precision	Sensitivity	Specificity	F-Score	Accuracy	Cohen's kappa
	0	587	142	1217	187	0.76	0.81	0.76	0.90	0.781		
	1	1217	187	587	142	0.90	0.87	0.90	0.76	0.881		
M1	Overall										0.85	0.66
	0	562	133	1232	206	0.73	0.81	0.73	0.90	0.768		
	1	1232	206	562	133	0.90	0.86	0.90	0.73	0.879		
M2	Overall										0.84	0.65
	0	1884	522	3966	735	0.72	0.78	0.72	0.88	0.750		
	1	3966	735	1884	522	0.88	0.84	0.88	0.72	0.863		
M3	Overall										0.82	0.61
	0	1889	448	4040	730	0.72	0.81	0.72	0.90	0.762		
	1	4040	730	1889	448	0.90	0.85	0.90	0.72	0.873		
M4	Overall										0.83	0.64
	0	1908	434	4054	711	0.73	0.81	0.73	0.90	0.769		
	1	4054	711	1908	434	0.90	0.85	0.90	0.73	0.876		
M5	Overall										0.84	0.65
	0	28496	3799	29967	5064	0.85	0.88	0.85	0.89	0.865		
	1	29967	5064	28496	3799	0.89	0.86	0.89	0.85	0.871		
M6	Overall										0.87	0.74
	0	2087	334	6650	838	0.71	0.86	0.71	0.95	0.781		
	1	6650	838	2087	334	0.95	0.89	0.95	0.71	0.919		
M7	Overall										0.88	0.70
	0	1947	310	6679	793	0.71	0.86	0.71	0.96	0.779		
	1	6679	793	1947	310	0.96	0.89	0.96	0.71	0.924		
M8	Overall										0.89	0.70
	0	2051	318	6672	832	0.71	0.87	0.71	0.95	0.781		
	1	6672	832	2051	318	0.95	0.89	0.95	0.71	0.921		
M9	Overall										0.88	0.70
	0	570	153	1206	204	0.74	0.79	0.74	0.89	0.762		
	1	1206	204	570	153	0.89	0.86	0.89	0.74	0.871		
M10	Overall										0.83	0.63

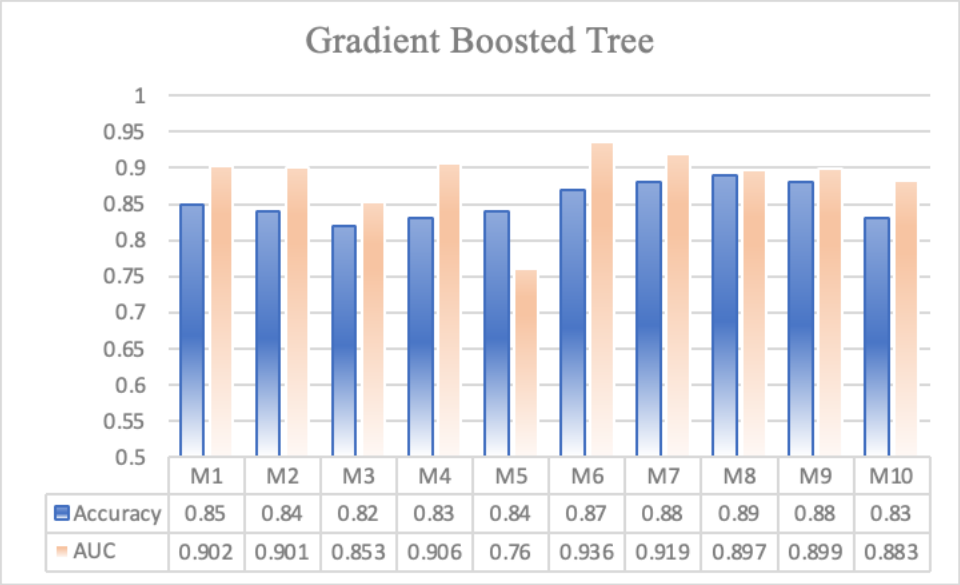


Figure 68. Gradient Boosted Trees accuracy and AUC comparison concerning change in hyperparameters and data pre-processing

Table 13

Backward Feature Elimination

Number of features	Accuracy	Removed Feature
63	0.894	All in
62	0.894	Q1_extrapaymentamount
61	0.894	Owner of pension product for woman
60	0.891	Yearly payment rate
59	0.895	Owner of pension product for children
58	0.892	Q3_payable amount
57	0.894	Yearly_extrapaid_amount
56	0.892	Critical disease product owner
55	0.894	Pension product owner
54	0.894	Q2 extra paid amount
53	0.898	Occupancy
52	0.891	Oldest product age
51	0.892	Payment by cash
………		

In the Gradient Boosted Tree backward feature elimination process, it was observed that removing the Q1_extrapaymentamount variable from the model had no effect on model performance while omitting the yearly pay rate variable resulted in a slight decrease in model performance. Furthermore, removing the Owner of pension product for children variable from the model resulted in a small improvement of 0.004 points in the model accuracy score. The feature selection process that maximizes the model accuracy score was selected as a result of 48 variables.

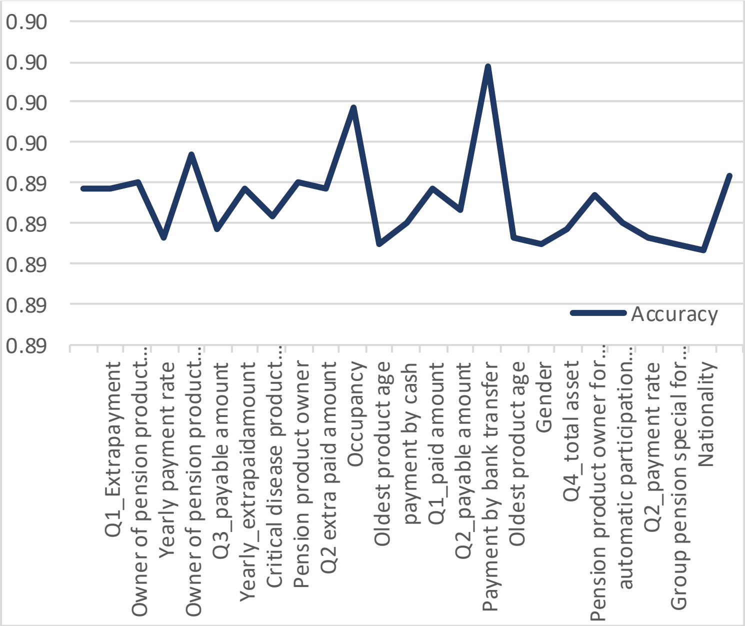


Figure 69. Backward feature elimination on Gradient Boosted Tree

In conclusion, it was observed that using XGboosting as a missing value handling option and giving importance to tree depth had a significant contribution to the model performance, while 10foldcross validation and normalization had little effect on the Gradient Boosted Tree. Additionally, it was observed that the bagging option: fraction of data to learn single model = 1 with replacement hyperparameter method significantly decreased the model performance. Furthermore, when modeling, KFold and feature elimination methods were found to require around 20 times more working time than the baseline model.

4.13 Naïve Bayes Application

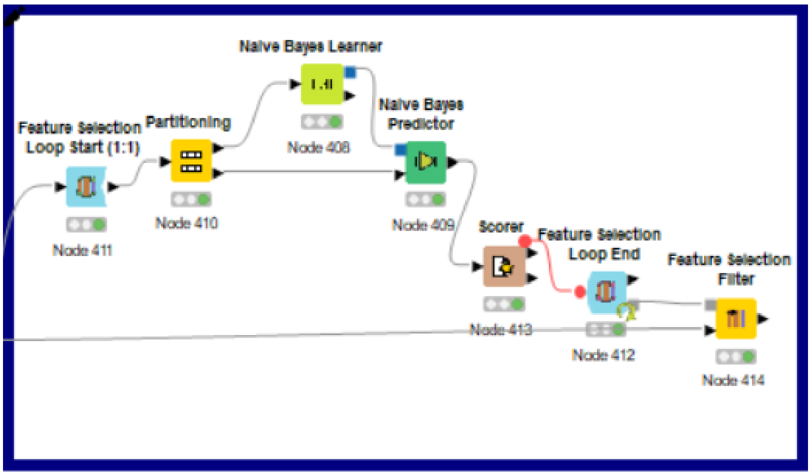


Figure 70. Naive Bayes structure

Table 14

Naive Bayes Performance

Model	Class	TP	FP	TN	FN	Recall	Precision	Sensitivity	Specificity	F-Score	Accuracy	AUC
M1	0	383	111	1236	403	0.49	0.78	0.49	0.92	0.60		
	1	1236	403	383	111	0.92	0.75	0.92	0.49	0.83		
	Overall										0.76	0.82
M2	0	604	261	1106	162	0.79	0.70	0.79	0.81	0.74		
	1	1106	162	604	261	0.81	0.87	0.81	0.79	0.84		
	Overall										0.80	0.85

In the first model using the Naive Bayes algorithm, the dataset was cleaned for outliers and missing values, and a train set of 70% was determined as the split method. This model resulted in an accuracy score of 0.76 and an AUC score of 0.82. In the second model, where dimension reduction was applied to variables with high correlation and the data set was normalized, there was a 0.1-point improvement in the accuracy score and a 0.03-point improvement in the AUC score.

4.14 Random Forest Application

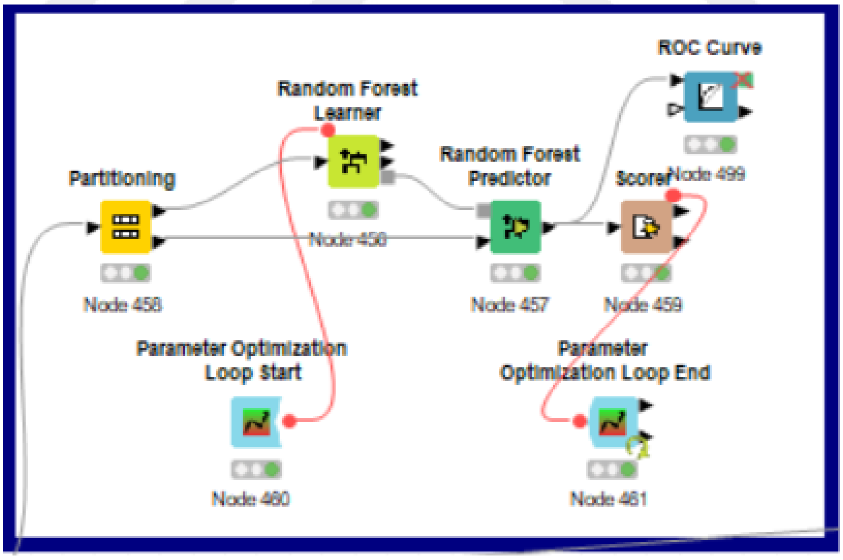


Figure 71. Random Forest structure

In the first model, data pre-processing steps, splitting the data into train and test sets, and hyperparameter tuning were conducted to assess their impact on the success of the random forest model. In this model, the test set was composed of 30% of the total data, and the train set was composed of the remaining 70%. Before proceeding with the modeling, cleaning procedures were performed for missing data and outlier values. To avoid bias in the data, it was deemed appropriate to not fill outlier values with the most common value or the mean. In the first model, the maximum tree depth was set to 20, and the tree split criterion was set to information gain.

Table 15

Applied Methods on Random Forest

Algorithm	Model	Normalization	10Fold Cross Validation	30/70 Test/Train	Missing Value Omitted Before Modeling	Outlier Handling	Hyperparameter opt.	
Random Forest Classifier	M1	-	-	+	+	+	Max depth:20	Tree option split criterion: information gain ratio
	M2	-	-	+	+	+	Max depth:20	Tree option split criterion: Gini index
	M3	-	+	-	+	+	Max depth:20	Tree option split criterion: information gain ratio
	M4	-	+	-	+	+	Max depth:20	Tree option split criterion: Gini index
	M5	-	+	-	+	+	Max depth:10	Tree option split criterion: Gini index
	M6	+	+	-	+	+	Max depth:10	Tree option split criterion: Gini index
	M7	-	-	+	-	+	Max depth:10	Tree option split criterion: Gini index
	M8	-	-	+	-	-	Max depth:10	Tree option split criterion: Gini index
	M9	-	+	-	-	+	Max depth:10	Tree option split criterion: Gini index

Table 16

Random Forest Model Performance

Model	Class	TP	FP	TN	FN	Recall	Precision	Sensitivity	Specificity	F-Score	Accuracy	Cohen's kappa
M1	0	573	160	1187	213	0.729	0.782	0.729	0.881	0.754		
	1	1187	213	573	160	0.881	0.848	0.881	0.729	0.864		
	Overall										0.825	0.619
M2	0	550	122	1225	236	0.700	0.818	0.700	0.909	0.754		
	1	1225	236	550	122	0.909	0.838	0.909	0.700	0.873		
	Overall										0.832	0.628
M3	0	186	51	400	73	0.718	0.785	0.718	0.887	0.750		
	1	400	73	186	51	0.887	0.846	0.887	0.718	0.866		
	Overall										0.825	0.616
M4	0	182	47	421	60	0.752	0.795	0.752	0.900	0.773		
	1	421	60	182	47	0.900	0.875	0.900	0.752	0.887		
	Overall										0.849	0.660
M5	0	194	36	413	67	0.743	0.843	0.743	0.920	0.790		
	1	413	67	194	36	0.920	0.860	0.920	0.743	0.889		
	Overall										0.855	0.680
M6	0	191	43	407	70	0.732	0.816	0.732	0.904	0.772		
	1	407	70	191	43	0.904	0.853	0.904	0.732	0.878		
	Overall										0.841	0.650
M7	0	641	109	1985	232	0.734	0.855	0.734	0.948	0.790		
	1	1985	232	641	109	0.948	0.895	0.948	0.734	0.921		
	Overall										0.885	0.711
M8	0	8643	1203	8927	1470	0.855	0.878	0.855	0.881	0.866		
	1	8927	1470	8643	1203	0.881	0.859	0.881	0.855	0.870		
	Overall										0.868	0.736
M9	0	2102	358	6632	816	0.720	0.854	0.720	0.949	0.782		
	1	6632	816	2102	358	0.949	0.890	0.949	0.720	0.919		
	Overall										0.882	0.701

In the second model, using the Gini index as the tree split criterion increased accuracy. In the third model, 10-fold cross-validation was used as the split method. This change was observed to increase the AUC value.

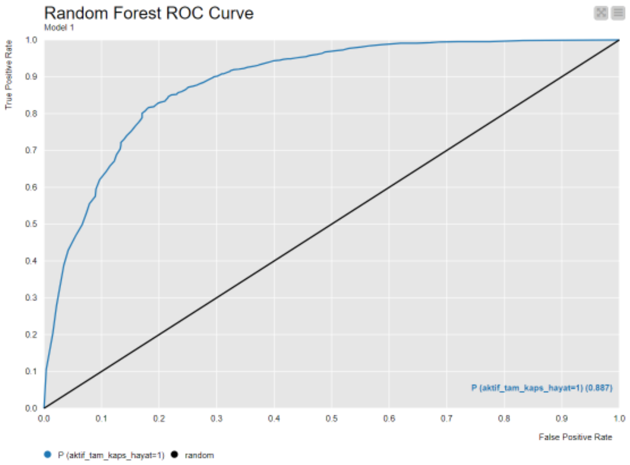


Figure 72. Random Forest ROC curve

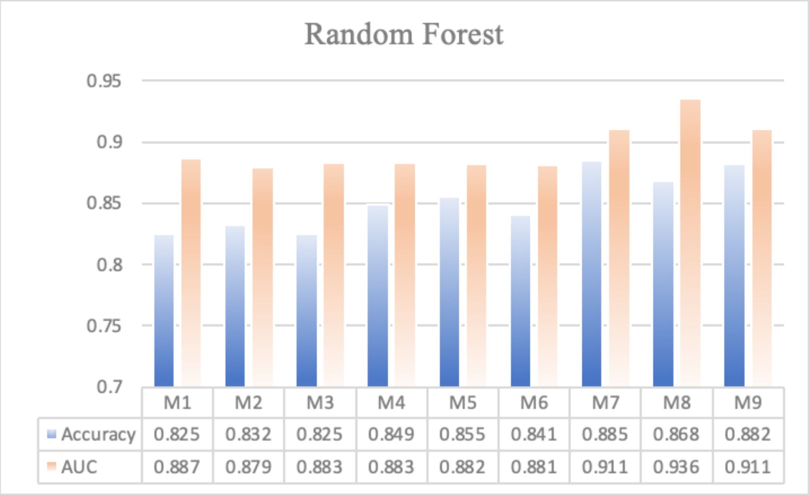


Figure 73. Random forest model performance

It was noted that normalization had a low impact on the success of the model. In the eighth model, the outlier cleaning and missing value handling steps were not performed, and the random forest algorithm was applied to the raw data directly. Although the maximum depth was low, an increase in AUC was observed. From the results of these models, it was concluded that using the Gini index instead of information gain for tree split and not cleaning missing values in the pre-processing step had a positive effect on model performance. It has been observed that the performance of a Random Forest model can be improved by utilizing the Gini index as the split criterion for decision trees, as opposed to the gain ratio. Additionally, omitting outliers and missing value handling during the data pre-processing phase has been shown to result in a statistically significant increase in the model's AUC, with a 0.1 point improvement being observed in this case.

Chapter 5

Discussions and Conclusions

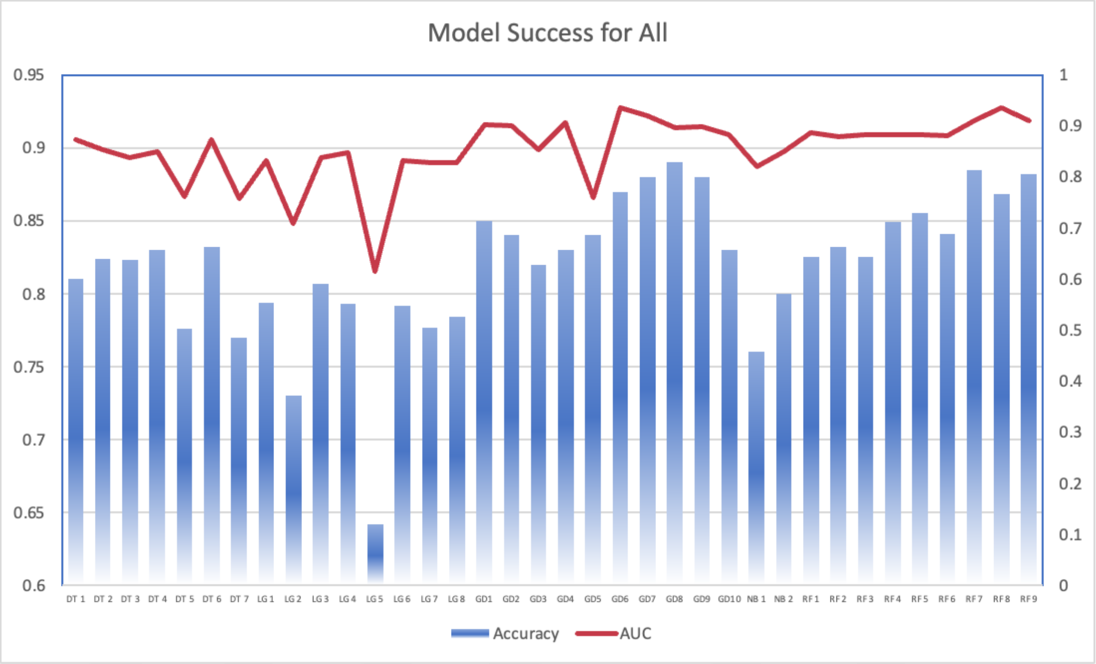


Figure 74. Model success for all

The aim of this study was to identify which of the company's current life insurance customers are suitable candidates for cross-selling additional life insurance products, in order to improve customer satisfaction through a personalized, needs-based offering. Customers who currently have a pension product have been considered as the positive class, and an equal number of customers who do not have the product have been selected randomly from the portfolio to form the negative set. In the data analysis, a significant difference was not observed between the two classes in terms of payment behavior other than regular payment (extra payment feature). However, it was observed that there were significant differences between the two groups in terms of characteristics such as the amount of assets in the customer's account, the customer's age, segment, customer lifetime, and ownership of other product groups. Credit card was found to be the most commonly preferred payment method according to the evaluation of the payment method type based on the accumulation amount in the customer's account. Customers who have been customers for less than 1,000 days were found to make more additional payments. The frequency of outliers in the assets of customers after 36 years of age increased. The age range with the highest account value was determined to be 40 years and older. Data on the amount customers need to pay annually and quarterly and their payment behavior are available in the data set. In the correlation analysis, it was found that payment behavior and the length of customer relationships had a high correlation of 0.9 for each quarter. In this research, tools such as WEKA, Bayesian-Lab, and SPSS were determined in the literature as the vehicle for conducting these evaluations. Based on the ability to easily integrate with Python programming language and handle large data, as well as ease of implementation, the KNIME Analytics Platform and the Python programming language were found to be appropriate for analysis and developed modeling on the platform. The literature review conducted for this study identified that predictive product offering studies are widely prevalent in the e-commerce industry, but there is a limited number of such studies in the life insurance sector. The research found that the life insurance sector typically offers a smaller range of products, with an average of 10 products available. In contrast, the e-commerce sector offers a much larger range of products, numbering in the millions. Based on this information, as well as the results of the literature review, the performance of various modeling algorithms were examined to predict which customers would be suitable candidates for cross-selling additional life insurance products. These algorithms included random forest, decision tree, gradient boosted tree, and naive Bayes. The effects of data pre-processing and hyperparameter tuning on the generalized model performance were also evaluated through the use of an experimental setup designed for this purpose. The decision tree algorithm was employed to solve a classification problem, and the performance of seven models was evaluated. Feature selection, train-test split methods, tree depth setting, and pruning using Minimum Description Length (MDL) were used to determine the best model, which was compared based on accuracy and AUC score. Results indicated that the forward feature selection method, combined with 10-fold cross-validation and pruning using MDL, achieved the highest AUC of 0.87 for the decision tree model. Although the normalization of the dataset had no significant effect on accuracy, it caused a 0.02 point decrease in AUC. Furthermore, it was observed that model performance decreased by 0.1 points when an optimal node and tree depth were not determined through hyperparameter optimization, especially when pruning was not applied. The independent variables of paid_amount and account_value_asset were found to carry sufficient information to reach an accuracy score of 80% through forward feature selection. In order to evaluate the impact of various techniques on the performance of a logistic regression model, a study was conducted in which normalization, epoch, missing value handling, dimension reduction and data split methods were compared. The results showed that the presence of missing values in the data renders logistic regression inapplicable. When a split method was utilized, in which 30% of the data was reserved for testing and 70% for training, an increase in the epoch value did not significantly enhance the model's performance. However, the application of 10-fold cross-validation revealed that a high epoch value resulted in an improvement in model performance. The accuracy of the model reached 0.79, while the AUC value was found to be 0.84. the performance of the Gradient Boosted Tree algorithm was compared using different parameters such as the depth of tree, learning rate, data split methods, and attribute sampling. One of the notable outcomes of this algorithm is its ability to produce high-performance models even when there are missing values, as demonstrated through the application of XGBoost. The effect of missing values on the AUC (area under the curve) was also examined using an ANOVA table, and it was found that there was a significant difference based on the p-value. By optimizing the max depth, applying 10-fold cross validation, and using XGBoost, a high score of 0.93 AUC was achieved with this algorithm. The increase in AUC score for the Naive Bayes algorithm was achieved by normalizing the data and using dimension reduction to eliminate highly correlated data. The AUC score achieved was 0.85. It was found that using the Gini index as the split criterion for the decision trees in a random forest algorithm resulted in a 0.1-point increase in the model's performance, as measured by the area under the curve (AUC) value. This improvement was observed when the data was not preprocessed to address outliers or missing values. The final AUC value obtained was 0.93. It was evaluated the performance of five different modeling techniques on a dataset. The XGBoost and Random Forest algorithms demonstrated the best performance, as measured by the area under the curve (AUC) value of 0.93. These two techniques outperformed the other three methods that were compared.

In this study, the potential for data leakage in payment behavior data was considered in the context of economic fluctuations in the market. The concentration of independent variables, such as occupation, in a single group in the dataset was also observed. Furthermore, it was known that during campaign periods, a certain group of customers may have been offered this product by the company, which could potentially lead to the formation of a biased dataset. These could potentially limit the performance of generalized models. Considering for further discussion, various artificial intelligence algorithms, including Generative Adversarial Networks (GANs), collaborative filtering, content-based filtering, deep learning, long-short term memory (LSTM), and low-rank matrix factorization, as well as Bayesian Networks, were analyzed for their potential use in product recommendation. Additionally, the possibility of using positive unlabelled learning, in which customers who do not have the product are treated as unlabelled data rather than as part of a negative dataset, can also be explored. As a further avenue for research, the potential use of life insurance cross-selling predictive models is suggested. These approaches may provide valuable insights for improving the accuracy and reliability of product recommendation systems in the future.

REFERENCES

Aguinis, H., Gottfredson, R. K., & Joo, H. (2013). Best-Practice Recommendations for Defining, Identifying, and Handling Outliers. In *Organizational Research Methods* (Vol. 16, Issue 2, pp. 270–301). SAGE Publications Inc. https://doi.org/10.1177/1094428112470848

al Shalabi, L., Shaaban, Z., & Kasasbeh, B. (2006). Data Mining: A Preprocessing Engine. *Journal of Computer Science, 2*(9), 735–739.

Albrecher, H., Bommier, A., Filipovic, D., Koch-Medina, P., Loisel, S., & Schmeiser, H. (2019). Insurance: Models, Digitalization, and Data Science. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3382125

Balabanović, M., & Shoham, Y. (1997). Content-Based, Collaborative Recommendation. *Communications of the ACM*, *40*(3), 66–72. https://doi.org/10.1145/245108.245124

Bannari Amman Institute of Technology, & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of the 2019 International Conference on Advances in Computing & Communication Engineering (ICACCE-2019) : 04-06 April 2019, Bannari Amman Institute of Technology, Sathyamangalam, India*.

Barot, S., Ronthal, A., Ramakrishnan, R., & Sun, J. (2022). *Migrating D&A Architectures to the Cloud: Roadmap*.

Bekker, J., & Davis, J. (2018). *Learning from positive and unlabeled data: a survey*. https://doi.org/10.1007/s10994-020-05877-5

Bouckaert, R. R. (2004). *Bayesian Network Classifiers in Weka*.

Chou, P. A., Lookabaugh, T., & Gray, R. M. (1989). Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling. *IEEE Transactions on Information Theory*, *35*(2), 299–315. https://doi.org/10.1109/18.32124

Clarke, C. L. A., Fuhr, N., Kando, N., Kraaij, W., & de Vries, A. P. (n.d.). *SIGIR '07 : 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval : July 23-27, 2007, Amsterdam, the Netherlands*.

Condliff, M. K., Lewis, D. D., Madigan, D., & Posse, C. (n.d.). *Bayesian Mixed-Effects Models for Recommender Systems*. http://cm.bell-labs.com/cm/ms/departments/sia

Dalian jiao tong da xue, & Institute of Electrical and Electronics Engineers. (n.d.-a). *Proceedings of IEEE 7th International Conference on Computer Science and Network Technology : ICCSNT 2019 : October 19-21, 2019, Dalian, China.*

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *ACM International Conference Proceeding Series*, *148*. https://doi.org/10.1145/1143844.1143874

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Global Insurance Report 2023: Reimagining life insurance. (n.d.).

Guo, J., Zhi, J., Fu, C. Q., Ran, H. L., & Zhao, X. L. (2014). Bayesian method applied to analyzing reliability of engineering machinery engine. *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2014*, 371–374. https://doi.org/10.1109/ITAIC.2014.7065073

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (1989). Applied Logistic Regression, 3rd Edition. *Wiley Series in Probability and Statistics*.

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–310. https://doi.org/10.1109/TKDE.2005.50

Jayalakshmi, T., & Santhakumaran, A. (2011). Statistical Normalization and Back Propagationfor Classification. *International Journal of Computer Theory and Engineering*, 89–93. https://doi.org/10.7763/ijcte.2011.v3.288

Kang, H. (2013). The prevention and handling of the missing data. In *Korean Journal of Anesthesiology* (Vol. 64, Issue 5, pp. 402–406). https://doi.org/10.4097/kjae.2013.64.5.402

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), 700–710. https://doi.org/10.1016/j.ijinfomgt.2016.04.013

Learning to Classify Texts Using Positive and Unlabeled Data. (n.d.).

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 179–186. https://doi.org/10.1109/icdm.2003.1250918

Mohanty, R., Ravi, V., & R. Patra, M. (2012). Classification of Web Services Using Bayesian Network. *Journal of Software Engineering and Applications*, *05*(04), 291–296. https://doi.org/10.4236/jsea.2012.54034

Murphy, K. P. (n.d.). *Naive Bayes classifiers*.

Naik, A., & Samant, L. (2016). Correlation Review of Classification Algorithm Using Data Mining Tool: WEKA, Rapidminer, Tanagra, Orange and Knime. *Procedia Computer Science*, *85*, 662–668. https://doi.org/10.1016/j.procs.2016.05.251

Noble, W. S. (2006). What is a support vector machine? In *NATURE BIOTECHNOLOGY* (Vol. 24). http://www.nature.com/naturebiotechnology

Nordhausen, K. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, Jerome Friedman. *International Statistical Review*, *77*(3). https://doi.org/10.1111/j.1751-5823.2009.00095_18.x

Pathak, B., Venkatesan, R., & Yin, F. (2010). Empirical Analysis of the Impact of Recommender Systems on Sales Empirical Analysis of the Impact of Recommender Systems on Sales Empirical Analysis of the Impact of Recommender Systems on Sales. *Article in Journal of Management Information Systems*. https://doi.org/10.2307/29780174

Patrick E. McKnight, Katherine M. McKnight, & Souraya Sidani. (n.d.). *Missing Data: A Gentle Introduction*.

presicion recall. (n.d.).

Priyadarshini, R. K., Bazila Banu, A., & Nagamani, T. (2019). Gradient Boosted Decision Tree based Classification for Recognizing Human Behavior. *Proceedings of the 2019 International Conference on Advances in Computing and Communication Engineering, ICACCE 2019*. https://doi.org/10.1109/ICACCE46606.2019.9080014

Qazi, M., Fung, G. M., Meissner, K. J., & Fontes, E. R. (2017). An insurance recommendation system using Bayesian networks. *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*, 274–278. https://doi.org/10.1145/3109859.3109907

Quinlan, J. R. (1986). Induction of Decision Trees. In *Machine Learning* (Vol. 1).

Rahman, M. S., Rivera, E., Khomh, F., Guéhéneuc, Y.-G., & Lehnert, B. (2019). *Machine Learning Software Engineering in Practice: An Industrial Case Study*. http://arxiv.org/abs/1906.07154

Rokach, L., & Maimon, O. (2006). Decision Trees. In *Data Mining and Knowledge Discovery Handbook* (pp. 165–192). Springer-Verlag. https://doi.org/10.1007/0-387-25465-x_9

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 3). Springer. https://doi.org/10.1007/s42979-021-00592-x

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719. https://doi.org/10.1142/S0218001409007326

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003a). *Algorithms for Large Scale Markov Blanket Discovery*. www.aaai.org

v Ferreira, J. E., Miranda, R. M., Figueiredo, A. F., Barbosa, J. P., & Brasil, E. M. (2016). Box-and-Whisker Plots Applied to Food Chemistry. *Journal of Chemical Education*, *93*(12). https://doi.org/10.1021/acs.jchemed.6b00300

Villarroel Ordenes, F., & Silipo, R. (2021). Machine learning for marketing on the KNIME Hub: The development of a live repository for marketing applications. *Journal of Business Research*, *137*, 393–410. https://doi.org/10.1016/j.jbusres.2021.08.036

Yang, X.-S., Dey, N., Joshi, A., & Institute of Electrical and Electronics Engineers. (n.d.-a). *Proceedings of the Third World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) : 30-31 July 2019, United Kingdom*.

Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07*. https://doi.org/10.1145/1277741.1277790

Zhou, M., Ding, Z., Tang, J., & Yin, D. (2018). Micro behaviors: A new perspective in E-commerce recommender systems. *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018-Febuary*, 727–735. https://doi.org/10.1145/3159652.315