UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Higher Polytechnic School of Alcoi

Machine Learning Tools for Customer Segmentation in Insurance

Master's Thesis

Master's Degree in Business Administration

AUTHOR: Osorio Muñoz, Celia

Tutor: Juan Pérez, Ángel Alejandro

Cotutor: Pérez Bernabeu, Elena

ACADEMIC YEAR: 2023/2024

ABSTRACT

This thesis explores the use of machine learning tools to optimize customer segmentation strategies in the insurance industry. Customer segmentation is a critical component of insurance marketing and customer relationship management. This research investigates the application of various machine learning algorithms and techniques to analyze insurance data and categorize customers into distinct segments. The core of this thesis involves a comprehensive examination of different machine learning approaches for customer segmentation. It includes a comparative analysis of clustering algorithms, supervised and unsupervised techniques, and the use of both structured and unstructured data sources. Real-world case studies and examples from the insurance industry illustrate the practical application of these methods. Ethical considerations and data privacy concerns are discussed in the context of customer segmentation using machine learning tools, ensuring that the research takes into account the broader implications and responsibilities involved. Furthermore, the research evaluates the impact of enhanced customer segmentation on insurance companies, exploring how it can lead to improved marketing strategies, more personalized services, and increased customer retention.

KEYWORDS

Machine Learning; Customer Segmentation; Insurance; Data Analysis

RESUMEN

Esta tesis explora el uso de herramientas de aprendizaje automático para optimizar las estrategias de segmentación de clientes en el sector de los seguros. La segmentación de clientes es un componente crítico del marketing de seguros y de la gestión de las relaciones con los clientes. Esta investigación estudia la aplicación de diversos algoritmos y técnicas de aprendizaje automático para analizar datos de seguros y clasificar a los clientes en segmentos distintos. El núcleo de esta tesis consiste en un examen exhaustivo de diferentes enfoques de aprendizaje automático para la segmentación de clientes. Incluye un análisis comparativo de algoritmos de agrupación, técnicas supervisadas y no supervisadas, y el uso de fuentes de datos estructuradas y no estructuradas. Estudios de casos reales y ejemplos del sector de los seguros ilustran la aplicación práctica de estos métodos. En el contexto de la segmentación de clientes mediante herramientas de aprendizaje automático, se examinan las consideraciones éticas y la protección de la privacidad de los datos, de modo que la investigación tenga en cuenta las implicaciones y responsabilidades más amplias que ello conlleva. Además, la investigación evalúa el impacto de una mejor segmentación de clientes en las compañías de seguros, explorando cómo puede conducir a mejores estrategias de marketing, servicios más personalizados y una mayor retención de clientes.

PALABRAS CLAVE

Aprendizaje automático; Segmentación de clientes; Seguros; Análisis de datos

RESUM

Aquesta tesi explora l'ús d'eines d'aprenentatge automàtic per a optimitzar les estratègies de segmentació de clients en el sector de les assegurances, component crític del màrqueting i la gestió de relacions amb els clients dins del mateix. Mitjançant la investigació de diversos algoritmes i tècniques d'aprenentatge automàtic, aquesta tesi analitza les dades d'assegurances per a categoritzar efectivament als clients en segments distints. Inclou una anàlisi comparativa d'algoritmes d'agrupament, tècniques d'aprenentatge supervisat i no supervisat, i l'ús de fonts de dades estructurades. Casos d'estudi reals i exemples del sector d'assegurances il·lustren l'aplicació pràctica d'aquests mètodes. Les consideracions ètiques i les preocupacions sobre la privacitat de les dades es discuteixen en el context de la segmentació de clients utilitzant eines d'aprenentatge automàtic, assegurant que la investigació tinga en compte les implicacions i responsabilitats més amples involucrades. A més, la tesi avalua l'impacte d'una millor segmentació de clients en les companyies d'assegurances, explorant com pot conduir a millors estratègies de màrqueting, serveis més personalitzats i una major retenció de clients.

PARAULES CLAU

Aprenentatge automàtic; Segmentació de clients; Assegurances; Anàlisi de dades

CONTENT INDEX

FIGURES INDEX

TABLES INDEX

ACRONYMS

AI: Artificial Intelligence

AUC: Area Under Curve

AAP: Average Annual Profit

BD: Big Data

DA: Data Analytics

EDA: Exploratory Data Analysis

FN: False Negatives

FP: False Positive

FPR: False Positive Rate

IM: Intelligent Marketing

IoT: Internet of Things

ML: Machine Learning

MSE: Mean Squared Error

NAs: Missing Values

ROC: Receiver Operating Characteristic

SDGs: Sustainable Development Goals

SML: Supervised Machine Learning

TN: True Negative

TP: True Positive

TPR: True Positive Rate

UML: Unsupervised Machine Learning

1. Introduction

The insurance sector, which is constantly looking for innovative ways to improve its operational efficiency (Garg & Garg, 2020) and customer service (Eckert et al., 2022), has the potential to fully adopt advanced technologies such as Big Data (BD) and Artificial Intelligence (AI). Because of this, this master's thesis focuses on the unexploited potential of these technologies, particularly in the field of customer segmentation.

1.1. Context

With the digitalization of society and economy, big amounts of data are constantly generated through the Internet of Things (IoT) (Hill et al., 2015). As a consequence, data plays a very important role in transforming modern businesses (Vassakis et al., 2018).

In today's markets, businesses of all sizes, from startups to multinational corporations, are pursuing a data-driven approach to secure a competitive advantage over their competitors (Vassakis et al., 2018). These businesses are focused on collecting and analyzing the data produced from their daily operations to obtain meaningful insights and crucial information for making informed, timely, and precise decisions on key business matters (Sivarajah et al., 2017) and, therefore, improving the whole decision-making process. Because of this, BD is becoming a topic of interest for both academics and business experts. BD and its tools, which include advanced analytics, machine learning (ML) algorithms, data visualization software, and data management systems (Singh & Singla, 2015), are being adopted by many companies from different sectors. Some of these sectors are health (Galetsi et al., 2020), entertainment (Hallur et al., 2021), energy (Zhou & Yang, 2016), transportation (Torre-Bastida et al., 2018), finance (Hussain, 2016) and insurance (Rana et al., 2022).

Based on this change towards more data-based strategies, there is a growing integration of AI in business to refine their decision-making processes in the digital era (Rajagopal et al., 2022). Furthermore, Enholm et al. (2022) highlight that this shift is instrumental in improving profitability, cost efficiency, and operational effectiveness.

From all the industries previously mentioned, this master's thesis will focus on insurance. This industry is a fundamental part of the global economy, providing financial security and risk management for individuals, businesses, and public organizations (Ray et al., 2020). It includes a variety of coverage types such as health, life, home, auto, property, casualty, and liability insurance, each tailored to manage specific risks, from medical emergencies, to property damage and vehicle accidents (Hanafy & Ming, 2021). Within this industry, the adoption of ML technologies, a subset of AI, is a transformative revolution on detecting fraud claims, as well as optimizing pricing strategies and risk assessment processes (Gupta et al., 2022). ML also offers potent tools for analyzing and interpreting complex datasets, allowing insurers to better understand their customer's behavior and segment them more effectively to customize their services accordingly (Jones & Sah, 2023).

Despite the benefits that AI and ML offer to the insurance industry, there is still considerable potential to be explored, particularly in the areas of customer segmentation (Abolmakarem et al., 2016). Current models, even though some are advanced, often do not fully exploit the patterns and trends that could be observed from deeper data analysis (Eluwole & Akande, 2022). This gap highlights an opportunity for further refining how insurance companies understand and categorize their customers. Additionally, the ability to dynamically classify customers based on evolving data sets is not fully used, which suggests that current tools and methods may be short in terms of exploiting AI's full predictive power (Luciano et al., 2023).

Based on these identified gaps in the application of AI and ML for customer segmentation within the insurance industry, the chosen master's thesis topic combines DA with strategic business management. As an MBA student deepening my expertise in ML, this topic not only aligns with my academic interests but also with my professional goals. My business education has taught me the importance of businesses having targeted and personalized communication for improving customer engagement. This has motivated me to approach the challenge of customer segmentation from both a strategic and technical perspective and develop a practical solution within the insurance industry. Therefore, this master's thesis wants to connect theoretical research with a practical implementation, improving how insurance companies understand and interact with their customers by correctly segmenting them.

1.2. Research Objectives

There are two types of insurance, life and non-life insurance, and this master's thesis will focus on the impact of AI in the non-life insurance industry, particularly auto insurance. Specifically, the starting point is to explore the use of ML in customer segmentation strategies within auto insurance.

Therefore, this master's thesis general objective is to apply SML techniques in an insurance company with the objective of improving their client segmentation process. This approach not only wants to improve the efficiency of the client segmentation process but also reduce the costs associated with poor segmentation and misclassification of clients. In order to this carry out this general objective, it has been specified in the following specific objectives:

- Study the applications of different SML techniques and algorithms within the insurance industry.
- Conduct a case study on an auto insurance company by proposing the development of a SML binary classification model that categorizes new clients into potential and non-potential based on their projected profit for the company.
- Perform a general Exploratory Data Analysis (EDA) on the dataset used in the case study, prior to the model development.
- Evaluate the performance of the developed model by analyzing key metrics such as Accuracy, F1 Score, Recall, ROC Curve, and AUC.
- Analyze, as well, the financial effects of the SML model in the company after its implementation by considering metrics such as Direct Costs, Avoided Costs and Opportunity Costs.

1.3. Relation of this Master's Thesis with the SDGs

The 2030 Agenda for Sustainable Development, as noted by the United Nations (United Nations, 2024), can be defined as a global plan adopted by all countries to eradicate poverty in all forms. For that, it includes 17 Sustainable Development Goals (SDGs) that aim for a sustainable development in an economic, social and environmental dimension.

The importance of this agenda is in its universal applicability, demanding collaborative efforts across borders, sectors, and disciplines (Shulla & Leal-Filho, 2023). It also emphasizes that sustainable development cannot be achieved in isolation but through an integrated approach that considers economic growth, social inclusion, and environmental protection. This vision makes the pursuit of the 2030 goals a main objective for the current generation (United Nations, 2024). After this definition and justification of the significance of the 2030 Agenda, Table 1 summarizes how the goals of this masters' thesis connect with its SDG objectives.

Table 1. Relation between this Master's Thesis and SDGs.

Sustainable Development Goals (SDGs)	**High**	**Medium**	**Low**	**Not applicable**
SDG 1. *No poverty*				**X**
SDG 2. *Zero hunger*				**X**
SDG 3. *Good health and well-being*			**X**	
SDG 4. *Quality education*			**X**	
SDG 5. *Gender equality*				**X**
SDG 6. *Clean water and sanitation*				**X**
SDG 7. *Affordable and clean energy*				**X**
SDG 8. *Decent work and economic growth*				**X**
SDG 9. *Industry, innovation, and infrastructure*	**X**			
SDG 10. *Reduced inequalities*				**X**
SDG 11. *Sustainable cities and communities*				**X**
SDG 12. *Responsible consumption and production*		**X**		**X**
SDG 13. *Climate action*				**X**
SDG 14. *Life below water*				**X**
SDG 15. *Life on land*				**X**
SDG 16. *Peace, justice, and strong institutions*				**X**
SDG 17. *Partnerships for the goals*				**X**

This master's thesis is closely aligned with several key SDGs, including 'Industry, Innovation, and Infrastructure (SDG 9)' and 'Responsible Consumption and Production (SDG 12)' with high and medium alignment, respectively. It also demonstrates a lower alignment with 'Good Health and Well-being (SDG 3)' and 'Quality Education (SDG 4)'. The specific contributions of this master's thesis to these four SDGs are detailed in Figure

1, highlighting how this research supports sustainable business practices and efficient resource utilization.

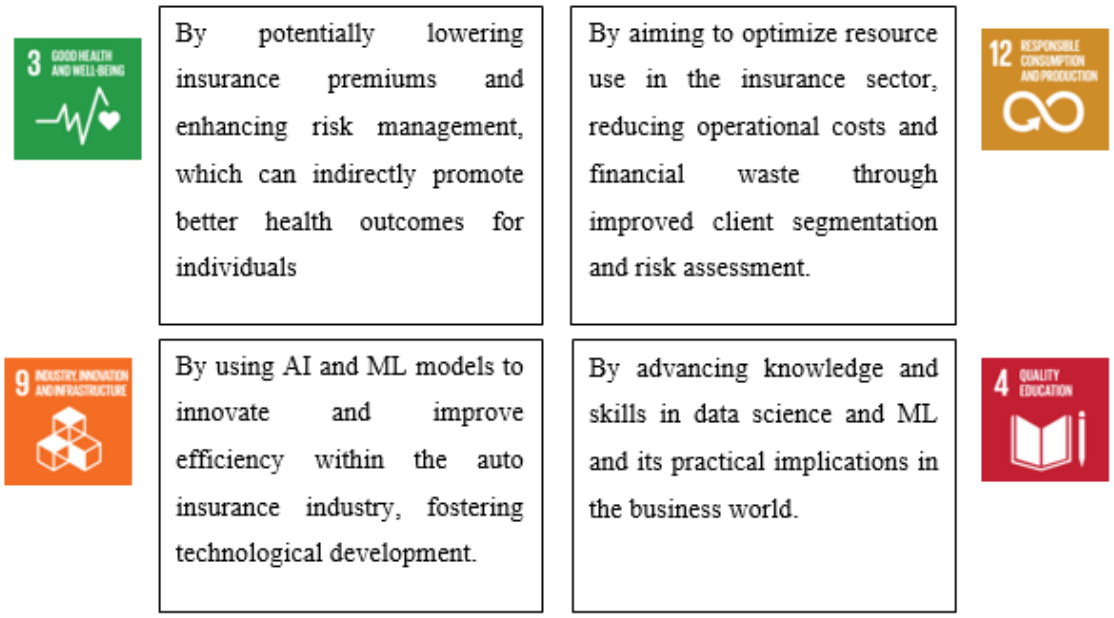

Figure 1. Specific contributions of the master's thesis to the SDGs.

1.4. Structural Overview

This master's thesis is structured to offer a comprehensive exploration of the application of SML tools for customer segmentation in the insurance industry. For this, it begins with an **Introduction** section that sets the starting point of the master's thesis by providing a context of the insurance sector and its tendency towards data-driven strategies. It also outlines the objectives of the thesis.

Following the introduction, the **Literature Review** is presented, focusing on existing research related to SML, customer segmentation, and their specific applications within the insurance industry.

The following section, **Methodology**, defines the methods considered to investigate the effectiveness of SML tools in enhancing customer segmentation. This includes a description of the statistical models and financial metrics considered.

The methodology leads to the **Case Study** section, which applies the discussed SML techniques to an insurance company. This section explores with a practical example how a company from this field can apply SML techniques to classify new clients more precisely and the consequent cost of misclassifying them.

Then, the **Conclusions and Future Work** section summarizes the research's main findings, contributions and limitations. It also highlights areas for future research, discussing further research steps in order to improve the proposed model.

Finally, the **References** section provides a list of all sources cited throughout the master's thesis. They help verify and sustain the presented information, as well as serve as an informative resource for future research projects in the field.

2. Literature Review

Through exploring the concepts of Marketing Intelligence (MI), customer segmentation and Supervised Machine Learning (SML), this section wants to provide an overview of how these concepts are changing the insurance industry as well as how businesses define their strategies.

2.1. Relevance and Advances in Marketing Intelligence and Segmentation

In a context of increasing customer expectations and demands for faster customer service, there is an expectation that companies understand their customers and quickly provide them appropriate services and recommendations of products (Guarda et al., 2012). In response, MI has become an important tool for businesses to collect, analyze, and use customer data effectively (Shah & Murthi, 2021). MI is a systematic and ongoing process that involves gathering, analyzing, and interpreting data about a company's internal and external environments in order to understand the dynamics, preferences, and behavior of clients, competitors, and the market at large (Chern et al., 2015).

As digital technologies advance, traditional marketing paradigms are being redefined, paving the way for a more data-driven approach to marketing and understanding and segmenting customers (Shah & Murthi, 2021) through MI. This shift in marketing practices is also sustained by authors Shah and Shay (2019), who point out that marketing is now focusing more on transformative technologies such AI, mixed reality, and blockchains. This evolution of technologies has allowed MI to use a variety of sophisticated tools and platforms to enhance customer understanding and engagement. Among these technologies, Big Data (BD) and Data Analytics (DA) stand out, allowing the treatment and interpretation of big amounts of data in order to identify patterns, trends, and consumer behaviors (Kumar & Reinartz, 2018). The ability to process and analyze data on such a large scale is fundamental for customer segmentation, campaign personalization, and customer experience optimization. Furthermore, to fully exploit the potential of big data in MI, it is essential to integrate disciplines such as data science, ML, text processing, audio processing, and video processing (Miklosik & Evans, 2020).

Because of this integration of advanced technologies in MI, the next step involves implementing predictive analytics to anticipate future consumer behaviors and market trends via AI and ML (Kotras, 2020). Some of the potential applications of these two technologies in Marketing Intelligence are highlighted by Haleem et al. (2022) in the following Figure 2.

Figure 2. AI uses in Marketing. Authors: Haleem et al. (2022).

By using ML algorithms, MI professionals can develop models that predict consumer responses to different marketing strategies (Dai & Wang, 2021). This not only enhances the precision of marketing campaigns but also allows companies to allocate resources more effectively, maximizing their return on investment (Zhang et al., 2022). However, for this to be effective, it is necessary for companies to invest in enhancing their marketing teams' skills in areas such as data science, AI, and data analytics, while also promoting a culture of experimentation and innovation within marketing departments (Thontirawong & Chinchanachokchai, 2021).

Moreover, AI in MI offers opportunities for more precise customer segmentation. By using AI and big data, companies can now segment their customers with more accuracy

than ever (Kasem et al., 2024). AI algorithms analyze big amounts of data collected from various sources like social media interactions, website visits, and purchase histories. This enables the identification of subtle patterns and trends that might be invisible to human analysts, allowing the creation of highly detailed customer groups (Turkmen, 2022). This advanced segmentation is not only about grouping customers based on traditional demographics like age or gender but also involves more dynamic factors such as buying behaviors, preferences, and even sentiments expressed across digital platforms (Kasem et al., 2024). For example, AI applied to customer segmentation can help distinguish between more influenced by brand loyalty, designing customized marketing strategies for each segment (Turkmen, 2022).

Furthermore, AI facilitates real-time segmentation, updating customer groups as new data becomes available, ensuring that marketing strategies remain relevant and aligned with changing customer behaviors and fast-paced market environments where consumer preferences can shift rapidly (Haleem et al., 2022).

2.2. Supervised Machine Learning Principles and Potential in Insurance

Within the insurance industry, this digital transformation is significant due to the integration of DA, AI and ML and the unprecedented opportunities that they offer to insurance companies (Yum et al., 2022).

ML is a subset of AI that focuses on developing algorithms that can learn from data, identify patterns, and make decisions with minimal human intervention (Nasteski, 2017). This enables machines to improve their performance and adapt to new environments through experience and exposure to more data. Specifically, as illustrated in Figure 3 by authors Shyam and Chakraborty (2021), SML involves training a model on a dataset where both the inputs and the desired outputs are known. Once trained, the model can apply these learned relationships to new, unseen data to predict outcomes effectively (Nasteski, 2017). Because of its characteristics, this type of learning is good for prediction and classification tasks because the model learns the relationship between the features of the data and their corresponding labels (Yum et al., 2022).

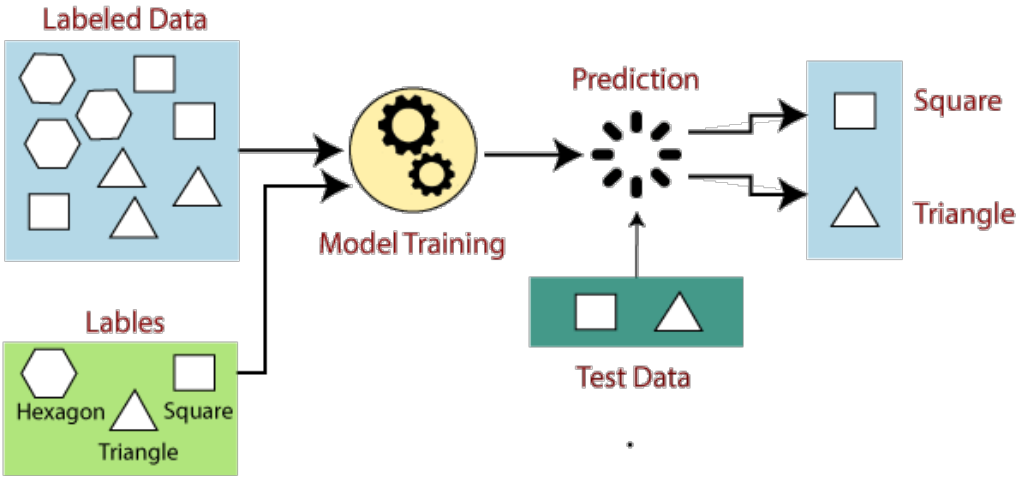

Figure 3. Workflow of a SML model. Authors: Shyam and Chakraborty (2021).

SML is one of the most used approaches of ML in insurance, especially for predicting risks, processing claims, and personalizing policies to individual needs (Debener et al., 2023). These and other applications are summarized in Figure 4.

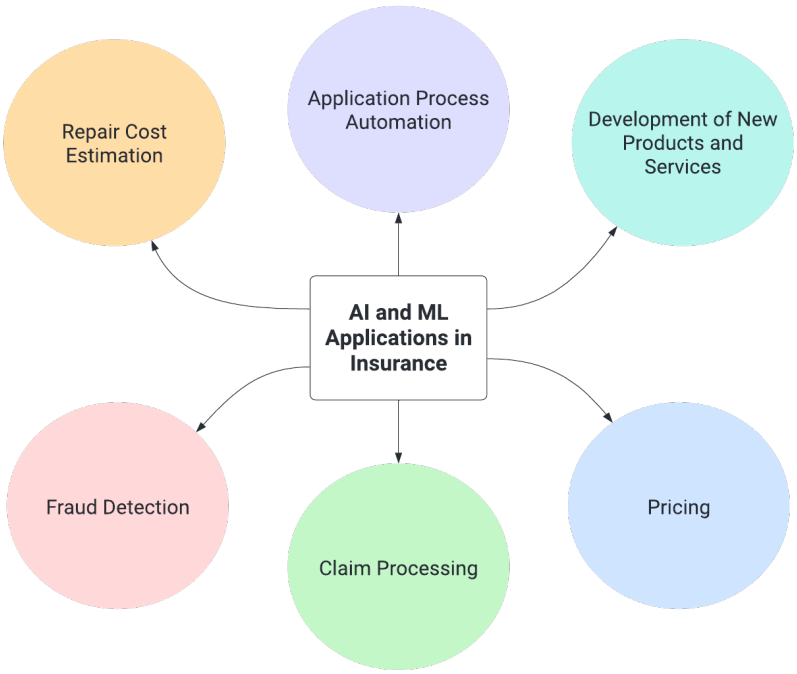

Figure 4. AI and ML Uses in Insurance.

Authors such as Scardovi (2017) suggest that the insurance sector might evolve toward an on-demand services model by offering insurance policies that are adapted in real-time to reflect customers' changing and specific behaviors. Through this type of "pay-as-you-go" and on-demand system, insurance companies could, for example, apply higher charges to individual clients whose behaviors suggest a higher risk (Scardovi, 2017). This would rely on SML models, trained on big datasets containing historical data on client behaviors, claims histories, and other relevant factors, to assess and adjust premiums based on real-time data dynamically. Such applications could rely on several SML algorithms such as logistic regression, decision trees, and ensemble methods like Random Forest and XGBoost, which are good at handling the complexities of large, variable datasets and making accurate predictions about future claim probabilities and customer risk profile (Boodhun & Jayabalan, 2018). These algorithms also facilitate personalized and fair pricing strategies (Hanafy & Ming, 2021).

SML is also useful for detecting fraud or high-risk patterns, allowing insurers to modify policy terms or take preventive actions proactively (Khodabandehlou & Zivari, 2017). This level of customization and adaptation not only helps insurers manage risk more effectively but also enables a more personalized customer experience, offering policies that reflect an individual's actual risk profile rather than a broad demographic segment.

The impact of digital transformation in the insurance sector is also noticeable in the introduction of innovative online marketing and sales strategies for customer engagement (Bohnert et al., 2019). In this case, using SML algorithms such as regression analysis, decision trees, and support vector machines to analyze customer interactions, can help insurers better predict insurance needs and fully understand and meet their clients' necessities (Scardovi, 2017).

2.3. Practical Applications of Supervised Machine Learning in Insurance

In this subsection, two applications of SML in the insurance industry will be explored. These include predicting the customer response, a binary classification task with the objective of forecasting if s customer will respond to an insurance renewal offer. Then, for customer segmentation, a multiclass classification task will be analyzed.

2.3.1. Dataset Description

The dataset used for these practical applications is obtained from Kaggle, a popular online platform where data scientists share datasets for research and development. This particular dataset consists of 9,134 rows of historical records of clients from an insurance company and 25 variables that represent distinct attributes of these clients.

Among the variables, there are identifiers such as Customer and Policy Numbers, alongside demographic information like State, Gender, Marital Status, and Education. Transactional data is also included with variables such as Number of Open Complaints, Number of Policies, Policy Type, Renew Offer Type, Sales Channel, Total Claim Amount, Vehicle Class, and Vehicle Size.

In terms of client interaction and behavior, the dataset includes variables such as Response (whether the customer responded to the offer), Coverage level (Basic, Extended, Premium), and Employment Status (Employed, Unemployed, Disabled, Medical Leave). Additionally, there is a series of encoded variables that translate qualitative attributes into quantifiable data, such as Location Code (Suburban, Rural, Urban), Vehicle Size (Small, Medium, Large), and Education levels (High School or Below, College, Bachelor, Master, Doctor).

2.3.2. Predicting Customer Response

The first application of SML in the insurance industry focuses on creating a model to predict whether clients will respond to insurance renewal offers. The "Response" column is used as the target variable, defining this as a binary classification problem where clients either respond or they don't. To address this, the XGBoost model is used for its efficiency in handling binary classification tasks like this one.

As illustrated in Figure 5, the model development phase starts with the division of the dataset into features ‘X’ and the target variable ‘y’. The features selected for this analysis include the variables 'State', 'Coverage', 'Education', 'Gender', and 'Income'. These are stored in ‘X’, while the target variable ‘Response’ is in ‘y’.

Next, the dataset is split into training and testing sets to evaluate the model's performance. 70% of the data is for training the model, which learns to classify the responses, and 30% for testing, where the model's predictive accuracy is evaluated with new unseen data. After completing the training phase, the model is then used to make predictions on the test set as illustrated in Figure 5.

```
# Split the dataset into features (X) and target variable (y)
selected_features = ['State', 'Coverage', 'Education', 'Gender', 'Income']
X = df[selected_features]
y = df["Response"]

# Encode the target variable
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Encode categorical variables in features
for column in X.select_dtypes(include=['object']).columns:
    X[column] = label_encoder.fit_transform(X[column])

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.3, random_state=42

# Initialize and train the XGBClassifier model
model = XGBClassifier(random_state=42)
model.fit(X_train, y_train)

# Perform predictions on the test set
predictions = model.predict(X_test)
```

Figure 5. SML Model Setup for Example 1.

Following the model training and predictions, the next step involves visualizing the confusion matrix for the test set predictions. This matrix is then shown in Figure 6, considering the encoding of "No" responses with label 0 and "Yes" with label 1.

This model accurately predicted true negatives (TN) with 2264 instances identified as non-responders. However, the model also recorded 143 true positives (TP), correctly identifying clients who responded affirmatively to the renewal offers, reflecting its capability to capture potential positive engagements.

On the other hand, the model had errors with 68 false positives (FP) and 266 false negatives (FN). The FP, where the model incorrectly anticipated positive responses, suggests a propensity to overestimate client engagement, leading to potential inefficiencies in marketing efforts. On the other hand, FN indicate missed opportunities, as these were actual positive responders misclassified as non-responders.

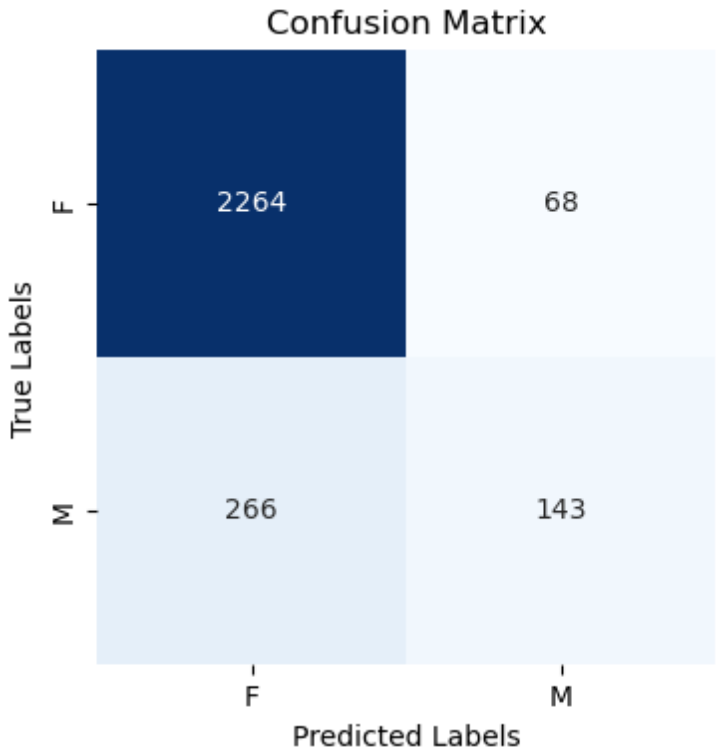

Figure 6. Confusion Matrix for Example 1.

Then, in Figure 7, the classification report offers a quantitative evaluation of the model's performance. This model achieves an overall accuracy of approximately 87.8%, indicating a high level of correct predictions. For class 0 (No), precision is high at 0.89, and recall is even stronger at 0.97, resulting in an F1-score of 0.93, which demonstrates that the model is effective for identifying non-responders. On the other hand, for class 1 (Yes), the metrics show that the precision drops to 0.68, while the recall is low at 0.35, and the F1-score is only 0.46. These figures highlight the model's difficulty in accurately predicting actual responders. Therefore, this is a focus for model improvements.

```
Accuracy: 0.878146661802262

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      2332
           1       0.68      0.35      0.46       409

    accuracy                           0.88      2741
   macro avg       0.79      0.66      0.70      2741
weighted avg       0.86      0.88      0.86      2741
```

Figure 7. Classification Report for Example 1.

Considering the results obtained, the XGBoost model is optimized using Randomized Search by doing a search for the best hyperparameters in Figure 8. The best parameters obtained from the search included 500 estimators, a maximum depth of 9, a learning rate of 0.2, a subsample rate of 0.7, and a column sample by tree of 0.6.

```python
# Define the hyperparameter search space for XGBoost
param_grid = {
    'n_estimators': [100, 200, 300, 400, 500],  # Number of gradient boosted trees
    'max_depth': [3, 5, 7, 9, 10],  # Maximum depth of a tree
    'learning_rate': [0.01, 0.05, 0.1, 0.2],  # Step size shrinkage used to prevent overfitting
    'subsample': [0.6, 0.7, 0.8, 0.9, 1.0],  # Subsample ratio of the training instances
    'colsample_bytree': [0.6, 0.7, 0.8, 0.9, 1.0]  # Subsample ratio of columns when constructing each tree
}

# Initialize RandomizedSearchCV
random_search = RandomizedSearchCV(model, param_distributions=param_grid, n_iter=100, cv=5, verbose=2,
                                   random_state=42, n_jobs=-1)

# Train the random search model
random_search.fit(X_train, y_train)

# Display the best hyperparameters found
print("Best Parameters:", random_search.best_params_)

# Make predictions with the optimized model
predictions = random_search.predict(X_test)
```

Figure 8. Search for Best Hyperparameters for Example 1.

As shown in Figure 9, these optimized settings led to a model accuracy of approximately 90%, an improvement from the earlier 87.8%. Also, the classification report reveals an increase on the precision levels from 0.87 to 0.93 for Class 0 (No) and from 0.68 to 0.71 for Class 1 (Yes).

```
Accuracy: 0.9000364830353885

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94      2332
           1       0.71      0.56      0.62       409

    accuracy                           0.90      2741
   macro avg       0.82      0.76      0.78      2741
weighted avg       0.89      0.90      0.89      2741
```

Figure 9. Classification Report with Best Hyperparameters.

2.3.3. Customer Segmentation Using Multiclass Classification

The second application of SML in the insurance industry focuses on applying the Random Forest algorithm for customer segmentation based on the target variable 'Policy Type' in the insurance industry. The aim is to classify customers into three policy types: 'Personal Auto', 'Corporate Auto', and 'Special Auto' based on their preferences on some key features. These features, which include 'State', 'Coverage', 'Education', 'EmploymentStatus', 'Gender', 'Income', 'Vehicle Class', and 'Vehicle Size', are selected for their significant potential to determine the type of policy a customer might choose.

Figure 10 illustrates the implementation steps for training the Random Forest model on the insurance dataset. The process begins with the preprocessing, where features such as 'State', 'Coverage', 'Education', 'EmploymentStatus', 'Gender', 'Income', 'Vehicle Class', and 'Vehicle Size' are extracted from the dataset to form the feature set ‘X’, while 'Policy Type' is set as the target variable ‘y’.

```
# Assuming 'Policy Type' is the target variable
target_column = 'Policy Type'
features = ['State', 'Coverage', 'Education', 'EmploymentStatus', 'Gender', 'Income', 'Vehicle Class', 'Vehicle Size']

# Dropping non-predictive columns and the target variable
X = df[features]
y = df[target_column]

# Encode categorical variables
label_encoders = {}
for column in X.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    X[column] = label_encoders[column].fit_transform(X[column])

# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Training the Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Predicting on the test set
y_pred = rf_classifier.predict(X_test)

# Evaluating the model
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Plotting the confusion matrix
plt.figure(figsize=(10, 8))
conf_matrix = confusion_matrix(y_test, y_pred)
class_names = y.unique()
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap="Blues", xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Figure 10. SML Model Setup for Example 2.

The dataset is then split into training and testing sets with a 70-30 ratio and the Random Forest Classifier is initialized with 100 estimators and trained on the training data. Additionally, a confusion matrix is plotted in to visualize the model's accuracy in correctly classifying the policy types. The results of this matrix can be seen in Figure 11 and it shows that the model correctly predicts 'Personal Auto' policies, with 1772 correct predictions against 208 instances incorrectly predicted as 'Corporate Auto' and 34 as 'Special Auto'. For 'Corporate Auto', the model correctly predicted 540 cases but misclassified 70 instances as 'Personal Auto' and 11 as 'Special Auto'. While having the least data, this 'Special Auto' category shows 91 correct predictions, with misclassifications including 13 instances as 'Corporate Auto' and 2 as 'Personal Auto'.

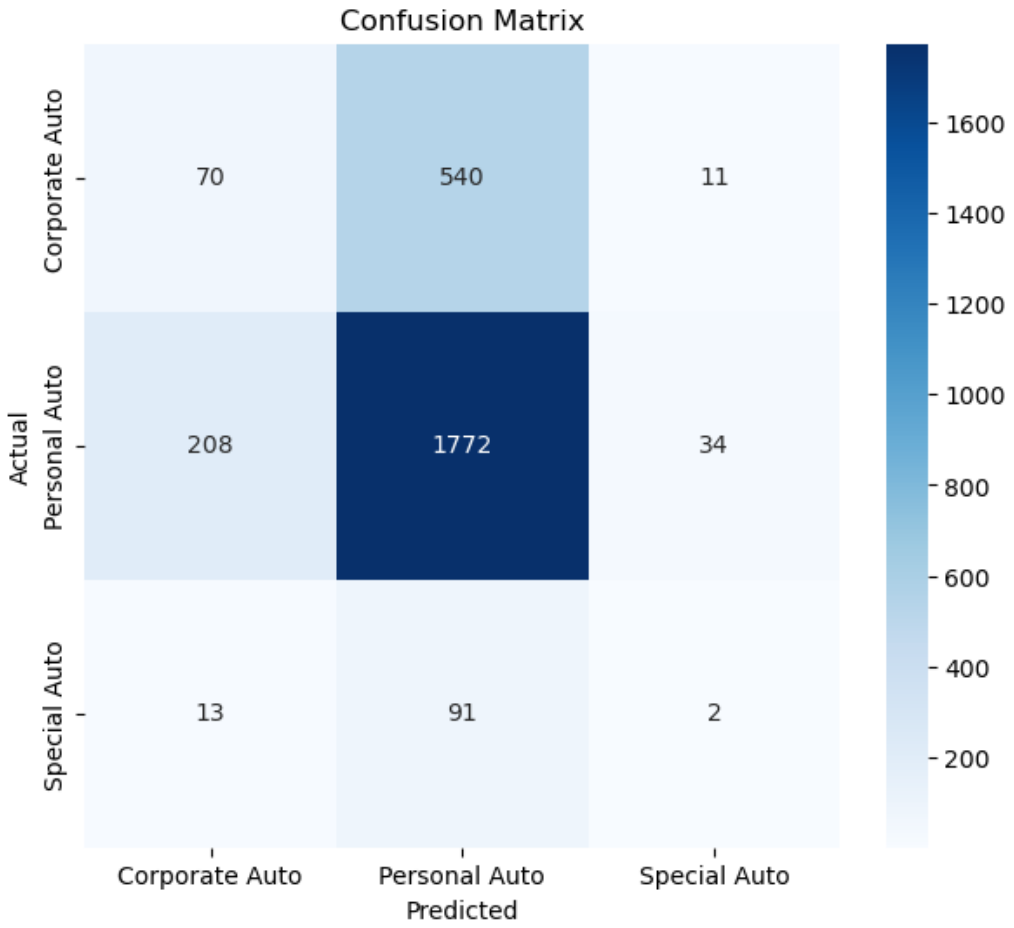

Figure 11. Confusion Matrix for Example 2.

Then, in Figure 12, the classification report highlights that the model performs well for the 'Personal Auto' category, demonstrating high precision and recall, which translates into an F1-score of 0.80. However, it struggles significantly with the 'Corporate Auto' and 'Special Auto' categories, which show much lower precision, recall, and F1-scores. The model's overall accuracy is 0.67, but the performance disparities across different classes suggest the need for further tuning.

```
Classification Report:
                precision    recall  f1-score   support

Corporate Auto       0.24      0.11      0.15       621
 Personal Auto       0.74      0.88      0.80      2014
  Special Auto       0.04      0.02      0.03       106

      accuracy                           0.67      2741
     macro avg       0.34      0.34      0.33      2741
  weighted avg       0.60      0.67      0.63      2741
```

Figure 12. Classification Report for Example 2.

Finally, in Figure 13, it is studied the importance of each feature to the model. This analysis reveals that 'Income' is the most important feature, contributing the most among all variables with an importance score of approximately 0.512. This indicates that a customer's income level is important in determining their preference for a policy type. Following 'Income', 'State' and 'Vehicle Class' have less impact, with importance scores of 0.109 and 0.096, respectively. On the other hand, 'Employment Status' and 'Gender' are the least influential from all the variables considered for the model, having the lowest scores.

```python
# Interpret feature importance
feature_importances = pd.DataFrame(rf_classifier.feature_importances_, index=X.columns,
                                   columns=['Importance']).sort_values('Importance', ascending=False)
print("\nFeature Importances:\n", feature_importances)
```

```
Feature Importances:
                  Importance
Income              0.512539
State               0.109702
Vehicle Class       0.096491
Education           0.089764
Coverage            0.060967
Vehicle Size        0.057597
EmploymentStatus    0.042450
Gender              0.030490
```

Figure 13. Feature Importance Analysis.

3. Methodology

Methodologically speaking, a dataset of automobile insurance clients is considered to develop a classification algorithm in Python, using libraries such as Pandas and Scikit-learn. It is important to note that the data has been anonymized to protect client confidentiality, with all variable names altered accordingly. This development process includes training, testing, and refining this algorithm to optimize its performance. For this purpose, the following statistical models are considered and compared:

- **Linear Regressor**, known for being straightforward and efficient in scenarios with direct linear relationships (Montgomery et al., 2021).
- **Lasso Regressor**, which incorporates L1 regularization to enhance model simplicity and interpretability by performing feature selection (Ranstam & Cook, 2018).
- **Ridge Regressor**, that is considered since it uses L2 regularization to reduce the impact of less critical features without eliminating them, therefore maintaining complexity while controlling for multicollinearity (Saleh et al., 2019).
- **Decision Tree Regressor** is evaluated for its ability to handle non-linear relationships and complex patterns (Czajkowski & Kretowski, 2016).
- **XGB Regressor**, that uses a gradient boosting framework to progressively improve accuracy by addressing previous errors, making it suitable for complex datasets (Avanijaa, 2021).
- **Huber Regressor**, which is useful for its robustness against outliers, considering a dual approach that balances the precision of least squares with the resilience of least absolute deviations (Sun et al., 2020).
- **LGBM Regressor** that works well handling large datasets with its histogram-based algorithm to manage continuous features, improving the processing speed and reducing memory demands (Yildirim et al., 2022).
- **CatBoost Regressor,** which offers similar advantages in improving the model accuracy and handling categorical data effectively (Heeb et al., 2022).

A financial analysis is also conducted in order to compare the original performance of the company with the numbers achieved through the implementation of the ML in the case study. With this, the objective is not just to determine the model's accuracy in classifying

potential and non-potential clients, but also financially quantify these classifications in for the insurance company. This analysis focuses on these four key financial metrics:

- **Earnings** that are calculated by subtracting the costs incurred from misclassifying potential clients as non-potential from the costs saved by correctly identifying potential clients.
- **Direct costs** reflecting losses from the incorrect classification of potential clients as non-potential.
- **Opportunity costs** which are related to incorrectly classifying a non-potential client as potential, leading to wasted resources and lost revenue opportunities.
- **Avoided costs** that are savings from correctly identifying non-potential clients and therefore avoiding unnecessary expenditures.

Furthermore, the effectiveness of the segmentation is evaluated using a set of statistical techniques and performance metrics such as Accuracy, F1 Score, Recall, and the Receiver Operating Characteristic (ROC) Curve and the Area Under the Curve (AUC) Score. Additionally, cross-validation is integrated into the evaluation process to improve the robustness of these results and mitigate any potential biases due to the randomness in the training data selection. This cross-validation method involves dividing the entire dataset into several subsets, then using different subsets to train and validate the model.

All in all, the workflow and phases for the development of the SML model proposed is visually summarized in Figure 14.

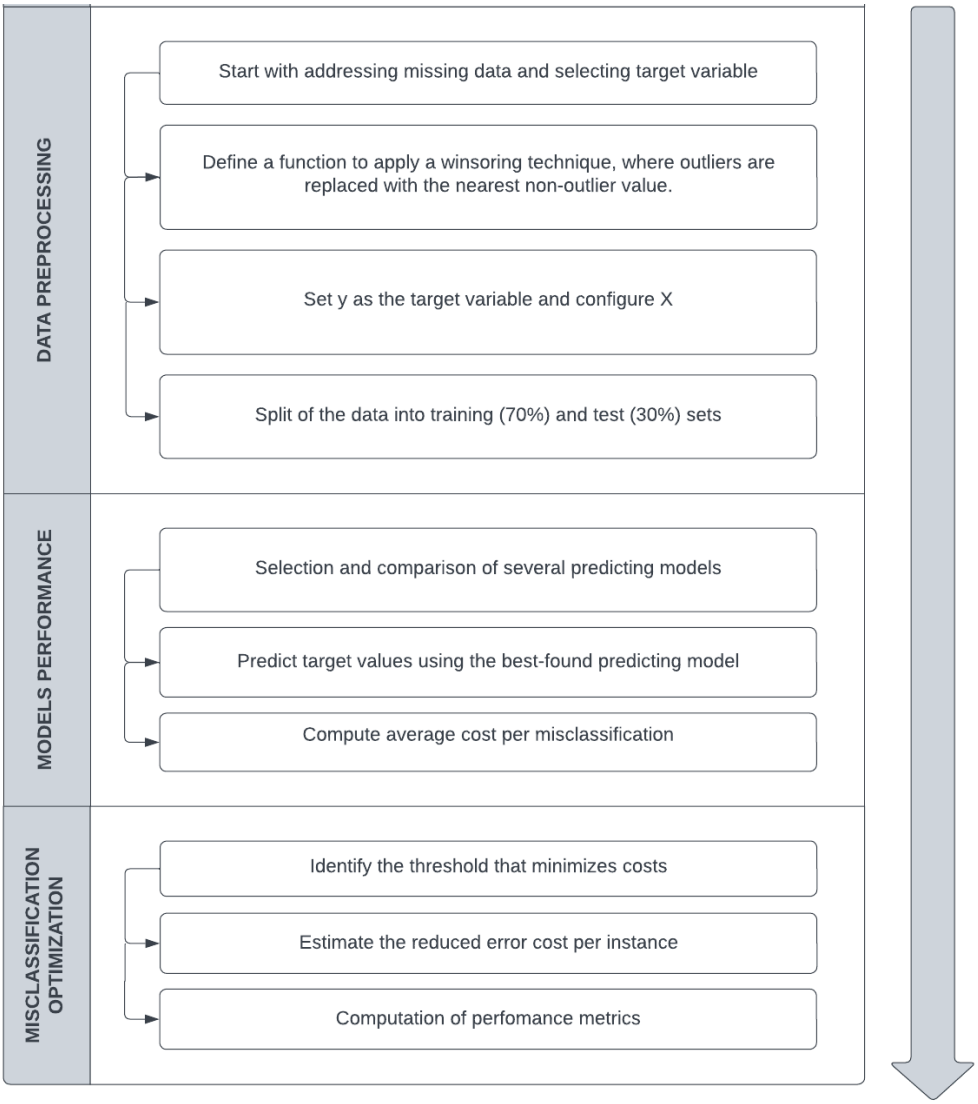

Figure 14. Workflow of Model Development.

The methodology also considers a detailed examination of academic literature, industry reports, and case studies of key topics like ML and MI in the insurance industry in order to set the theoretical base for conducting a practical case study application forward in this master's thesis. Some of the sources consulted for this include external secondary sources such as academic and business reports, publications in specialized insurance and information technology journals related to the key topics previously mentioned and official websites of institutions associated with the insurance industry, among others. This secondary information is gathered using specialized academic and scientific databases, such as Google Scholar and ScienceDirect.

4. Case Study

After exploring relevant literature and key terminology in the section of Literature Review, this master's thesis will now focus on a practical implementation through a case study for an insurance company.

This case study aims to develop a SML model for client segmentation, categorizing new clients as potential (class 0) or non-potential (class 1) based on their projected profitability. Additionally, the study includes an analysis of misclassification costs to quantify the financial impact of errors in the classification process. The development of this SML model is organized into three main phases: Data Preprocessing, Model Performance and Misclassification Cost Optimization.

4.1. Data Preprocessing

The Data Preprocessing phase includes a Exploratory Data Analysis (EDA) as the initial step for preparing the dataset considered in the model development. EDA begins with the dataset being loaded from a CSV file into Jupyter Notebooks using Python. At this point, essential libraries are imported to facilitate data manipulation and visualization. These libraries include pandas for data manipulation and analysis, matplotlib and pyplot for plotting, seaborn for enhanced statistical visualization, numpy for numerical computing with arrays and matrices, and the re library for string manipulation.

Next, the dimensions of the dataset are determined using the shape attribute, which indicates the number of rows and columns present. Also, the column names are extracted and printed, now labeled with generic identifiers such as 'C01', 'C02', 'C16', among others. To ensure compatibility with different data processing and ML tools, Unicode characters are standardized, non-alphanumeric characters are removed, and column names are shortened if they exceed a certain length.

The data structure is then analyzed in terms of data types, which reveals a dataset composed mainly of numerical data, with significant columns of type float64 (decimal numbers) and int64 (integers), along with one column classified as an object (text or mixed types). Then, an examination of missing values (NAs) confirms the absence of

NAs across the dataset, simplifying the preprocessing steps by eliminating the need for preliminary data cleaning.

Following the observation that there are no NAs in the dataset during the EDA, the data preprocessing continues with the target variable being defined as Average Annual Profit (AAP). This variable focuses on explaining the financial performance of the policyholders. After defining this target variable, the next step involves computing basic descriptive statistics to understand its distribution within the dataset such as the mean, median, standard deviation, and range of values. The results of this analysis suggest that the mean tends toward a slight negative, indicating an overall loss in the dataset. A substantial standard deviation indicates that there is a significant variability in AAP. This high degree of variability, along with the negative average, indicates the possibility of having outliers that may be dragging the mean downwards, presenting a more negative trend than would typically be observed if these extreme values were less pronounced or absent.

In order to further study this possibility, a boxplot of the distribution of AAP is created in Figure 15. This actually demonstrates the presence of outliers that are influencing the average of the target variable among others, contributing to the overall negative mean value observed in the descriptive statistics of the numerical variables.

```
# Boxplot of the target variable
def plot_boxplot(var):
    plt.figure(figsize=(8, 1))
    plt.boxplot(df[var], vert=False)
    plt.title('Boxplot of ' + var)
    plt.xlabel('Value')
    plt.grid(True)
    plt.show()

plot_boxplot(target)
```

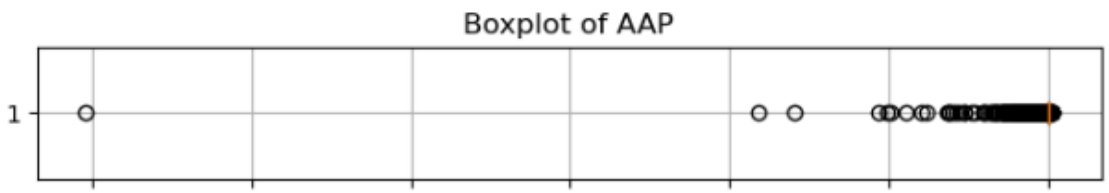

Figure 15. Distribution of AAP.

Therefore, the winsorizing method is used in Figure 16. This technique involves modifying outliers by replacing them with the closest boundary values determined based on the data's interquartile range. Quartiles divide the data into four equal parts after it has been sorted in ascending order; specifically, the first quartile (Q1) is the 25th percentile, and the third quartile (Q3) is the 75th percentile of the data. Boundaries for outliers are defined as 1.5 times the IQR below Q1 and above Q3.

```python
# Define a function to replace outliers with the closest non-outlier value (winsoring technique)
def modify_outliers(var):
    Q1 = var.quantile(0.25)
    Q3 = var.quantile(0.75)
    IQR = Q3 - Q1
    lb = Q1 - 1.5 * IQR
    ub = Q3 + 1.5 * IQR
    var[var < lb] = lb
    var[var > ub] = ub
    return var

# Iterate over numerical columns and replace outliers
if replace_outliers == True:
    for column in df.select_dtypes(include=[np.number]).columns:
        if column != 'AAP_original': # keep outliers in this column for future use
            df[column] = modify_outliers(df[column])
    print("Outliers have been replaced by their closest bound.")
else: print("No outliers have been replaced.")
```

Figure 16. Winsoring Technique for Outliers Management.

After implementing these measures to address outliers, the distribution of the target variable has improved, as shown in Figure 17. Now, the distribution is more compacted and centered around the median.

```python
# Boxplot of target after winsoring
plot_boxplot(target)
```

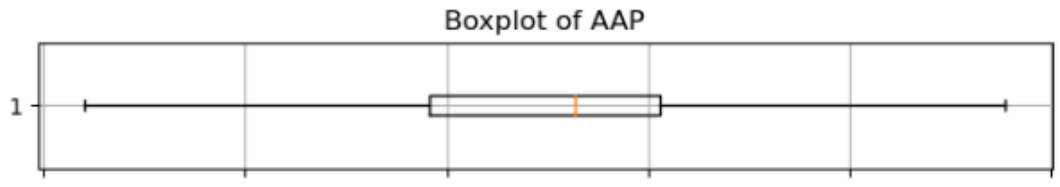

Figure 17. Boxplot of AAP After Winsoring.

The last step in the data preprocessing involves defining the predictive features (X) and the target variable (y). Here, the target variable 'y' is assigned to AAP. 'X' is then constructed by selecting only the numerical columns from the DataFrame while excluding the original target variable and other specific variables that are highly correlated. Then, the dataset is divided into training and testing sets to validate the model's result. Here, 70% of the data is used for training and 30% for testing.

4.2. Model Performance

In the Model Performance phase, the focus shifts to exploring and selecting the most accurately predictive model after the Data Preprocessing phase. For this, the first step is defining a list of several models to evaluate as shown in Figure 18. This list includes several models such as Linear Regression, Decision Tree Regressor, XGBoost Regressor, and CatBoost Regressor, among others, since their respective advantages meet different aspects of the data and the prediction objectives.

```
# Define list of regression models to try
models = [
    LinearRegression(),
    Lasso(),
    Ridge(),
    DecisionTreeRegressor(),
    XGBRegressor(),
    HuberRegressor(),  # More robust against outliers
    LGBMRegressor(verbose=-1),  # Added LightGBM Regressor
    CatBoostRegressor(silent=True)  # Added CatBoost Regressor
]

print("The list of regression models to try has been set.")
```

Figure 18. Definition of Models to Evaluate.

Then, as shown in Figure 19, the model performs a statistical analysis using the selected models in order to compare them and select the best-performing one. For this, each model is fitted with the training data, and its performance is evaluated through both training and testing datasets. The Mean Squared Error (MSE) and the Adjusted R-squared are

calculated in order to study the models' accuracy and fit. MSE measures the average squared difference between the estimated values and actual values. This is a commonly used risk metric that helps to evaluate the extent of error in a model. A lower value of MSE indicates a better fit. Then, the Adjusted R-squared, on the other hand, measures the proportion of variance for a dependent variable that's explained by an independent variable or variables in a regression model. Unlike the regular R-squared, the adjusted version also takes into account the number of predictors in the model, which helps to provide a more accurate measure of the model's explanatory power, especially when comparing models with different numbers of predictors.

```
# Using train data, perform a regression analysis with the selected models
def perform_regression(model):
    print(type(model).__name__)
    model.fit(X_train, y_train) # use train data for the fit
    predictions_train = model.predict(X_train)
    predictions_test = model.predict(X_test)
    mse_train = mean_squared_error(y_train, predictions_train)
    mse_test = mean_squared_error(y_test, predictions_test)
    r_sq_train = r2_score(y_train, predictions_train)
    r_sq_test = r2_score(y_test, predictions_test)
    n_train = len(X_train)
    n_test = len(X_test)
    p = len(X_train.columns) # columns are the same in both train and test datasets
    adj_r_sq_train = 1 - (1 - r_sq_train) * (n_train - 1) / (n_train - p - 1)
    adj_r_sq_test = 1 - (1 - r_sq_test) * (n_test - 1) / (n_test - p - 1)
    print("MSE for train data:", round(mse_train,2))
    print("MSE for test data:", round(mse_test,2))
    print("R-sq(adj) for train data:", round(adj_r_sq_train,2))
    print("R-sq(adj) for test data:", round(adj_r_sq_test,2), "\n")
    return adj_r_sq_test

# Select the model with the highest r-sq(adj) for the test data
best_adj_rsq_test = 0
for model in models:
    adj_r_sq_test = perform_regression(model)
    if adj_r_sq_test >= best_adj_rsq_test:
        best_adj_rsq_test = adj_r_sq_test
        best_model = model
```

Figure 19. Models Evaluation.

The results of the Adjusted R-square values for both training and test sets of each model show different effectiveness levels, as shown in Table 2.

Table 2. Comparison of Models Performance.

Model	R-sq(adj) for train data	R-sq(adj) for test data
Linear Regressor	0.62	0.62
Lasso	0.62	0.62
Ridge	0.62	0.62
Decision Tree Regressor	1.00	0.52
XGB Regressor	0.89	0.79
Huber Regressor	0.54	0.54
LGBM Regressor	0.83	0.79
CatBoost Regressor	0.86	0.80

Models like Linear Regression, Lasso, and Ridge show a consistent performance, indicating a good generalization of the model without having a significant overfitting. In contrast, while perfect in training, the Decision Tree Regressor, shows a decline in test performance, which suggests overfitting. Advanced ensemble methods such as XGBoost Regressor, LGBM Regressor, and CatBoost Regressor have a high performance, correctly balancing the complex data interactions with the generalization of the model with new unseen data. Also, the Huber Regressor maintains a moderate performance across both train and test datasets. Finally, the CatBoost Regressor is then selected as the chosen model primarily due to its good performance on the test set, where it achieved an adjusted R-squared value of 0.80. This metric indicates a high predictive accuracy, meaning that the model explains 80% of the variability in the target variable, making it a superior choice for this analysis in comparison to the results obtained from the rest of the models.

After selecting the best-performing model, the next step involves defining the true class values for the dataset. A threshold is established to categorize the dataset into potential new clients and non-potential new clients, using a binary classification system. This process assigns one class to values above the threshold and another class to values at or below it. This classification method is consistently applied to both the training and testing sets. Following this, the target values are predicted using the previously selected model, CatBoost in both the training and testing datasets.

After predicting these target values, the following step is identifying an optimal threshold that maximizes the model's ability to distinguish between potential clients and non-potential clients, measured by the AUC score. The AUC score evaluates how well a model distinguishes between the two classes, class 0 (potential clients), and class 1 (non-potential clients). For this, in Figure 20, random threshold values are generated between Q1 and Q3 using a triangular distribution, focusing on the median..

```
# Find the threshold value that 'optimizes' the selected score using the train set
print("Computations made using the train dataset...")
score_to_max = roc_auc_score # f1_score, recall_score, precision_score, accuracy_score, roc_auc

Q1 = y.quantile(0.25)
Q3 = y.quantile(0.75)
best_threshold = y.quantile(0.50)
best_score = -1

max_time = 30 # loop duration in seconds
start_time = time.time()
elapsed = 0
while elapsed < max_time:
    threshold = np.random.triangular(Q1, best_threshold, Q3)
    class_predicted_train = [0 if pred > threshold else 1 for pred in predictions_train]
    score = score_to_max(class_true_train, class_predicted_train)
    if score > best_score:
        best_score = score
        best_threshold = threshold
    elapsed = time.time() - start_time

print("Best threshold for " + score_to_max.__name__ + ": " + str(round(best_threshold,2)))
print("Best value for " + score_to_max.__name__ + ": " + str(round(best_score,2)))
```

Figure 20. Optimizing Roc_auc_score Threshold.

Then, a timed loop, limited to 30 seconds, facilitates multiple iterations to explore these thresholds. Each iteration predicts classes based on the current threshold, and the performance score is evaluated. If a score exceeds the current best, it's updated along with the threshold. This process culminates in a 'best threshold' of 17.11 and a score of 0.91 for AUC. This threshold favors a more conservative classification approach because, by selecting a threshold of 17.11, the model more frequently classifies individuals as class 1 (non-potential clients), thereby decreasing the likelihood of FP and minimizing revenue losses from mistakenly overlooked opportunities.

Then, the model's predictive accuracy is evaluated on the training dataset by determining the misclassification costs and the distribution of class pair instances. Therefore, as shown in Figure 21, it is calculated the misclassification costs and counts for each combination of predicted and actual class labels.

```python
# Compute missclassification cost in the train set
def calc_missclassification_cost_dictionary(class_predicted, class_true, df):
    dictionary = {k: {"cost": 0, "count": 0} for k in [(i, j) for i in range(2) for j in range(2)]}
    for i, (pred, true) in enumerate(zip(class_predicted, class_true)):
        dictionary[(pred,true)]["count"]+=1
        dictionary[(pred,true)]["cost"]+=(df.values[i])
    return dictionary

# Predict the class based on the best threshold
class_predicted_train = [0 if pred > best_threshold else 1 for pred in predictions_train]
dictionary = calc_missclassification_cost_dictionary(class_predicted_train, class_true_train, y_train)
print("Computations made using the train dataset...")
print(dictionary)

cost_matrix = [[0, 0], [0, 0]]
count_matrix = [[0, 0], [0, 0]]

# Fill the matrices with the data
for (row, col), values in dictionary.items():
    cost_matrix[row][col] = format(values['cost'], ",.2f")
    count_matrix[row][col] = format(values['count'], ",.2f")

# Convert matrices into DataFrames
df_cost = pd.DataFrame(cost_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])
df_count = pd.DataFrame(count_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])

# Print the DataFrames
print("Cost Matrix:")
print(df_cost)
print("\nCount Matrix:")
print(df_count)

for (row, col), values in dictionary.items():
    cost_matrix[row][col] = values['cost']
    count_matrix[row][col] = values['count']

# Convert matrices into DataFrames
df_cost = pd.DataFrame(cost_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])
df_count = pd.DataFrame(count_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])

# Calculate totals
total_cost = df_cost.to_numpy().sum()
total_count = df_count.to_numpy().sum()

# Convertimos a porcentajes
df_cost_percentage = round((df_cost / total_cost) * 100, 2)
df_count_percentage = round((df_count / total_count) * 100, 2)

# Print the percentage DataFrames
print("Cost Matrix (%):")
print(df_cost_percentage)

print("\nCount Matrix (%):")
print(df_count_percentage)
```

Figure 21. Misclassification Cost Dictionary in Train Set.

Once the misclassification costs and counts are compiled, they are organized into two separate matrices: one for costs and another for counts. Each matrix element corresponds to a specific combination of predicted and true class values, which helps in noticing areas where the model performs well and where it needs improvements. The resulting percentage matrices are displayed in Table 3 and Table 4.

Table 3. Cost Matrix in % for Train Set.

	Actual Potential Client	Actual Non-Potential Client
Predicted Potential Client	67.62 %	1.65 %
Predicted Non-Potential Client	1.59 %	29.14 %

Table 4. Count Matrix in % for Train Set.

	Actual Potential Client	Actual Non-Potential Client
Predicted Potential Client	68.22 %	2.77 %
Predicted Non-Potential Client	5.03 %	23.98 %

Then, the model predicts once again these classes and their respective cost and count matrices using the same code as in Figure 21 but applied on the test dataset instead of training. The results of these two matrices are displayed in Table 5 and Table 6.

Table 5. Cost Matrix for Test Set in %.

	Actual Potential Client	Actual Non-potential Client
Predicted Potential Client	68.06 %	2.50 %
Predicted Non-Potential Client	2.20 %	27.24 %

Table 6. Count Matrix for Test Set in %.

	Actual Potential Client	Actual Non-potential Client
Predicted Potential Client	67.88 %	3.45 %
Predicted Non-Potential Client	5.77 %	22.90 %

Following this computation of misclassification costs and counts in the test set, the next step is the generation of a confusion matrix that categorizes predictions into four key outcomes: TP, TN, FP and FN.

In this context, a TP outcome occurs when the model correctly identifies a potential client, affirming the model's effectiveness in recognizing valuable and potential clients. A TN, on the other hand, occurs when the model correctly identifies a non-potential client as such, reflecting the model's ability to accurately identify these cases that should not proceed in terms of relation with these types of clients and, therefore, saving up resources from the company. Furthermore, FP represents instances where the model wrongly labels non-potential clients as potential, leading to wasted resources and efforts. Also, FN describes scenarios where potential clients are misclassified as non-potential, which can result in missed opportunities and lost potential incomes.

This matrix indicates a larger number of instances where potential clients were correctly identified (TP) compared to accurately predicted non-potential clients (TN). It also registered fewer instances where potential clients were incorrectly labeled as non-potential (FN) than instances where non-potential clients were mistakenly identified as potential (FP). This indicates that the model is more effective at identifying potential clients, but still has some errors with FP and FN, with FN occurring more frequently than FP. Then, a classification report is generated in Figure 22 in order to evaluate the model's performance in the test dataset.

Classification report for test dataset:

	precision	recall	f1-score
0	0.95	0.92	0.94
1	0.80	0.87	0.83
accuracy			0.91
macro avg	0.88	0.90	0.88
weighted avg	0.91	0.91	0.91

Figure 22. Classification Report for Test Set.

Precision is particularly significant as it measures the accuracy of the predictions for potential and non-potential clients, indicating how many of the model's positive classifications were correct. Recall studies the model's capability to capture all relevant instances of each class, essentially measuring how many actual positives were not missed. The f1-score, a mean of precision and recall, is an indicator of the overall accuracy. For Class 0 (potential clients), the model shows very high precision and recall, effectively identifying true potential clients. Class 1 (non-potential clients) has slightly lower precision but high recall, demonstrating a good performance in identifying most non-potential clients with some errors. The overall accuracy is 91%, therefore predicting a high percentage of cases correctly.

Then, the next step involves computing the ROC curve and the AUC for the test dataset in order to measure the model's performance across all possible classification thresholds, as shown in Figure 23.

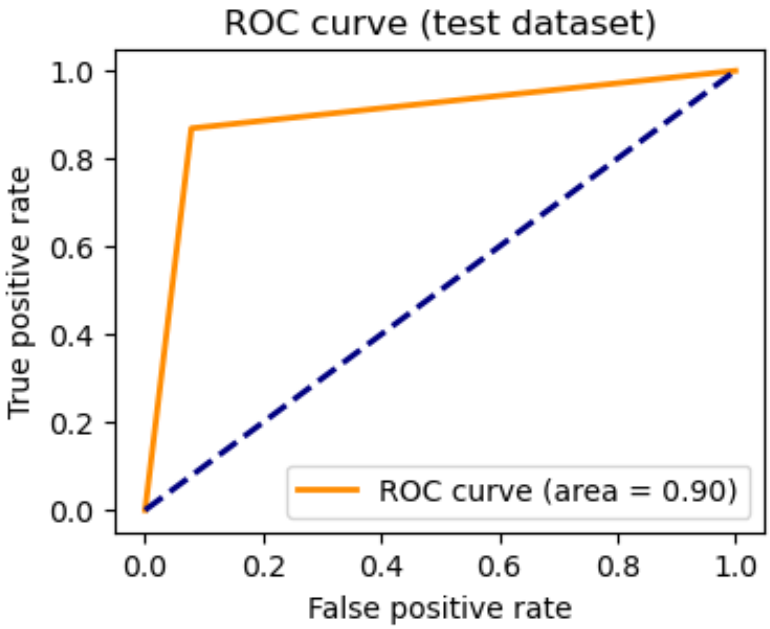

Figure 23. ROC and AUC for Test Set.

The ROC curve is a graphical representation that plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The TPR is also known as sensitivity or recall and it measures the proportion of actual positives that are correctly identified as such. The FPR, on the other hand, measures the proportion of actual negatives that are falsely identified as positives. The curve provides a tool to select the

threshold that best balances sensitivity and specificity based on the needs of a particular application.

As for AUC, it is a single value that summarizes the overall ability of the test to discriminate between the positive and negative classes irrespective of any particular threshold. An AUC of 0.5 suggests no discriminative ability (random guessing), while an AUC of 1.0 indicates perfect discrimination. In this case, the AUC, noted as 0.90 in the plot, indicates a high level of discriminative ability.

The final step in this second phase of developing the SML model involves a financial analysis to compare the insurance company's performance before and after implementing the proposed model.

In the financial analysis conducted before the implementation of the new model, the insurance company's earnings reflected their total revenue from successful transactions and operations. The original costs, which included losses from misclassifications and other operational inefficiencies, represented a significant portion of their expenses.

Following the introduction of the SML model, the updated financial metrics were analyzed. The implementation of this model resulted in a 67.93% increase in earnings, highlighting the model's positive impact on the company's financial performance. Also, the model has facilitated the identification of opportunity costs and avoided costs. Opportunity costs, identified as (1,0) where the first figure is the predicted class and the second is the actual class, represent the lost opportunities when a potential client is incorrectly classified as non-potential. Avoided costs, quantified as (1,1), represent the money saved by the company through correct classifications of non-potential clients. This breakdown of costs allows for a clearer differentiation of costs according to the type of classification error.

4.3. Optimization of Misclassification Cost

In this third phase, the primary goal is to mitigate the effects of errors in the model by fine-tuning its parameters, reducing this way both financial and operational costs resulting from incorrect predictions, therefore enhancing the model's overall effectiveness and efficiency.

After identifying the total costs for the insurance company using the test set, the objective is to optimize the decision threshold to reduce these costs. By refining the previously defined threshold of 17.11, the model not only becomes more accurate but also significantly more cost-efficient. To achieve this, as shown in Figure 24, the model uses a random search method, systematically testing different threshold values between specified quantiles of the data distribution. During each iteration, the algorithm makes predictions on the train set based on the current threshold being tested. It classifies the data into the previously defined classes (0 or 1) and calculates the misclassification cost using a predefined cost function, which considers the financial impact of each misclassification from the AAP target variable. The algorithm identifies and saves the threshold that results in the lowest cost by comparing the total misclassification cost at each step.

```
print("Computations made using the train dataset...")
Q1 = y.quantile(0.25)
Q3 = y.quantile(0.75)
best_threshold = y.quantile(0.50)
best_total_cost = float('inf')
best_total_misclassified = 0

max_time = 30  # maximum duration for the loop in seconds
start_time = time.time()
elapsed = 0
while elapsed < max_time:
    threshold = np.random.triangular(Q1, best_threshold, Q3)
    # Predict classes based on the threshold
    class_predicted_train = [0 if pred > threshold else 1 for pred in predictions_train]
    tc, tm, acc = calc_missclassification_cost(class_predicted_train, class_true_train)
    # Update best threshold and total cost if the current total cost is lower
    if tc < best_total_cost:
        best_total_cost = tc
        best_threshold = threshold
        best_total_missclassified = tm
        best_avg_cost_per_customer = acc
    elapsed = time.time() - start_time
print(f"Best threshold for minimizing total cost: {round(best_threshold, 2)}")
print(f"Total cost: {round(best_total_cost, 2)}")
print(f"Total misclassified observations: {best_total_missclassified}")
print(f"Average cost per misclassified customer: {round(best_avg_cost_per_customer, 2)}")
```

Figure 24. Optimization of Misclassification Threshold Using Train Set.

In this instance, the optimal threshold is found to be 34,78 using the train set, which ensures that there is no overfitting. Then, the model does the same process for the test dataset to detail the financial repercussions of each prediction error during the test. This function pairs each predicted value with the actual class, updating the dictionary to track the total costs and frequency of misclassifications for each possible outcome (0,0), (0,1), (1,0), and (1,1), being the first value the predicted and the second one the real.

Next, as shown in Figure 25, the model processes the test set to generate a dictionary that details the misclassification costs and counts.

```
# Calculate average cost per misclassified customer in the test set
print("Computations made for the test dataset...")
class_predicted_test = [0 if pred > best_threshold else 1 for pred in predictions_test]
dictionary_test = calc_missclassification_cost_dictionary(class_predicted_test, class_true_test, y_test)

cost_matrix = [[0, 0], [0, 0]]
count_matrix = [[0, 0], [0, 0]]

# Fill the matrices with the data
for (row, col), values in dictionary_test.items():
    cost_matrix[row][col] = round(values['cost'], 2)
    count_matrix[row][col] = round(values['count'], 2)

# Convert matrices into DataFrames
df_cost = pd.DataFrame(cost_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])
df_count = pd.DataFrame(count_matrix, columns=['Column 0', 'Column 1'], index=['Row 0', 'Row 1'])

# Print the DataFrames
print("Cost Matrix:")
print(df_cost)
print("\nCount Matrix:")
print(df_count)

# Calculate totals
total_cost = df_cost.abs().to_numpy().sum()
total_count = df_count.to_numpy().sum()

# Convert to percentage
df_cost_percentage = round((df_cost.abs() / total_cost) * 100, 2)  # Ensure percentages are based on absolute costs
df_count_percentage = (df_count / total_count) * 100

# Print the percentage DataFrames
print("\nCost Matrix (%):")
print(df_cost_percentage)

print("\nCount Matrix (%):")
print(df_count_percentage)
```

Figure 25. Dictionary for Optimized Misclassification Costs.

Following this, the total misclassification cost and total number of misclassifications are transformed into percentages to provide a proportional view of misclassification impacts. The results of these cost and count matrices are summarized in Table 7 and Table 8, respectively.

Table 7. Cost Matrix in % for Test Set for Optimizing Misclassification Cost.

	Actual Potential Client	Actual Non-Potential Client
Predicted Potential Client	67.31 %	2.14 %
Predicted Non-Potential Client	2.95 %	27.60 %

Table 8. Count Matrix in % for Test Set for Optimizing Misclassification Cost.

	Actual Potential Client	Actual Non-Potential Client
Predicted Potential Client	66.01 %	2.84 %
Predicted Non-Potential Client	7.64 %	23.51 %

Then, the confusion matrix shown in Figure 26 is generated to visually analyze the model's performance on the test dataset.

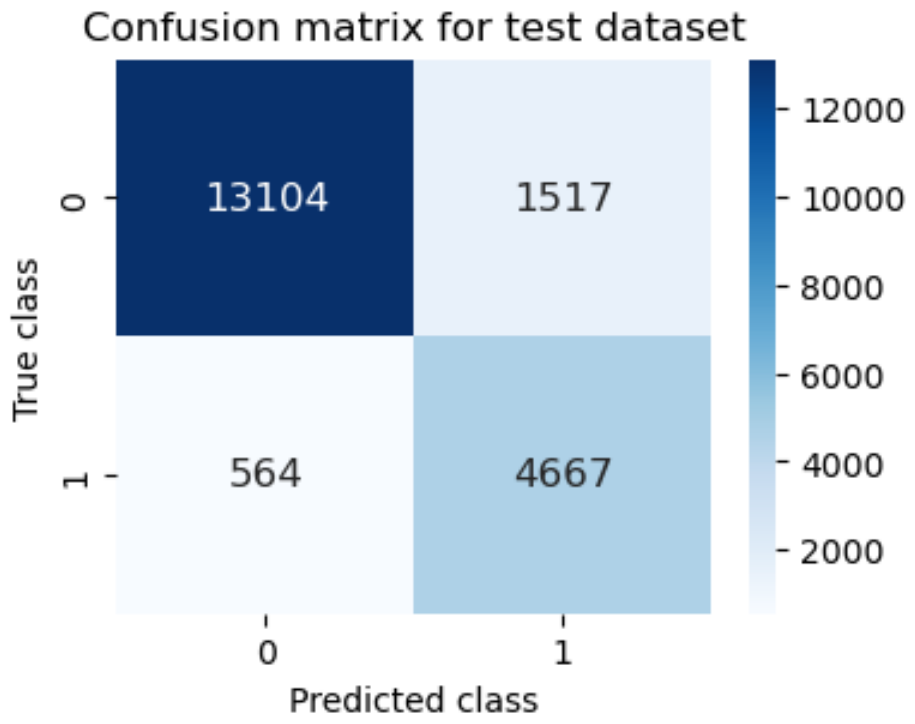

Figure 26. Confusion Matrix for Test.

This confusion matrix shows that there are significantly more true positives (TP) and true negatives (TN) compared to false positives (FP) and false negatives (FN). However, there are more false positives than false negatives, indicating that the model incorrectly labels potential clients as non-potential more frequently than it misses identifying non-potential clients. A classification report was generated as illustrated in Figure 27 to further study the model's performance.

Classification report for test dataset:

	precision	recall	f1-score
0	0.96	0.90	0.93
1	0.75	0.89	0.82
accuracy			0.90
macro avg	0.86	0.89	0.87
weighted avg	0.91	0.90	0.90

Figure 27. Classification Report for Test after Optimization.

This report shows a precision of 0.96 for potential clients and 0.75 for non-potential clients, with recall rates of 0.90 and 0.89 respectively. The F1-scores, 0.93 for potential clients and 0.82 for non-potential clients, indicate a robust balance between precision and recall, confirming the model's effectiveness in distinguishing between the two classes. The high precision for potential clients suggests that the model is very effective at correctly identifying true potential clients, minimizing false positives. Conversely, the lower precision for non-potential clients indicates a higher rate of false positives, where non-potential clients are incorrectly labeled as potential. The recall rates for both classes are quite balanced, with 0.90 for potential clients and 0.89 for non-potential clients, showing the model's proficiency in capturing the majority of actual instances in both categories, though it is slightly more effective at identifying true potential clients. Furthermore, the higher F1-score for potential clients (0.93) compared to non-potential clients (0.82) suggests that the model performs better overall in predicting potential clients.

After this, the ROC curve and its associated AUC are computed to evaluate the performance of the model. The ROC curve plots the FPR on the x axis and the TPR on the y axis and provides a graphical representation of the trade-off between sensitivity and specificity. The AUC, with a value of 0.89 as seen in Figure 28, indicates a high degree of separability between the potential and non-potential client classes.

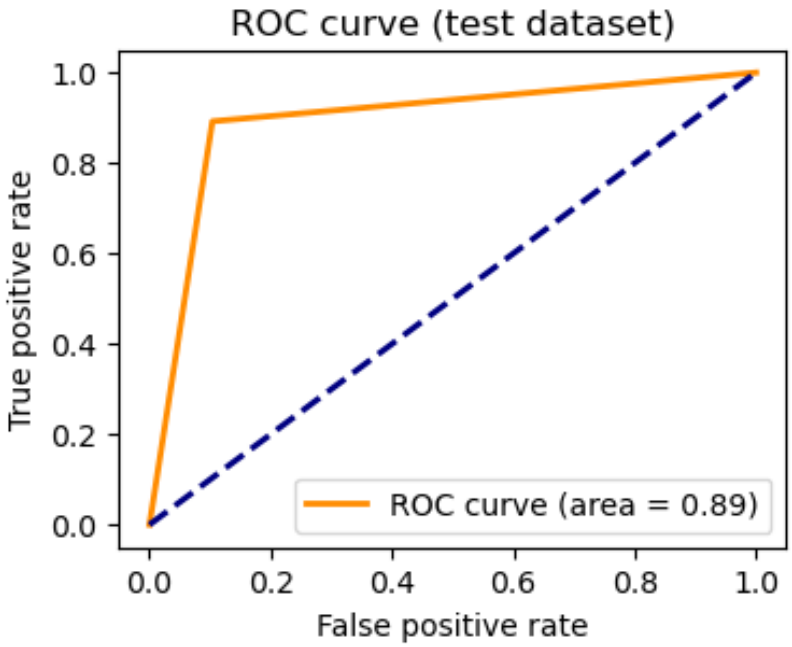

Figure 28. ROC Curve for Test after Optimization.

After the evaluation of the ROC curve and its AUC, the next step involves analyzing the financial outcomes of the optimization strategies, specifically targeting the optimization of misclassification costs. Comparing the figures from the proposed SML model before and after the optimization of the misclassification cost threshold, there is a significant improvement in financial metrics. Although there is a slight decrease of 1.10% in earnings, the performance of the company has improved by reducing total operational costs by 14.41% and opportunity costs by 33.93%. Additionally, the optimization strategies lead to the avoidance of 1.32% in potential costs.

4.4. Discussion of Results

This subsection evaluates the efficacy of the applied ML techniques and the overall performance of the model, as well as its economic impact on the insurance company featured in the case study, particularly in terms of the costs associated with misclassifying clients.

4.4.1. Analysis of Model Performance and Results

The performance of the SML model was studied through a series of tests to determine its accuracy and effectiveness in classifying potential and non-potential clients. The analysis focused on two distinct phases of the SML model development: 1) before Misclassification Cost Optimization, 2) after Misclassification Cost Optimization.

The model initially demonstrated its capacity to recognize opportunities for client engagement by correctly identifying a large number of potential clients. However, it also mistakenly identified several non-potential clients as potential, indicating a precision shortfall that could lead to unnecessary expenditure on ineffective marketing efforts. Additionally, the model effectively prevented resource wastage by correctly identifying many non-potential clients, though it overlooked a number of potential clients, pointing to lost opportunities.

Following the optimization of misclassification costs, there were some notable changes in the model's performance. The number of correctly identified potential clients decreased by 2.75%, still reflecting the model’s ability to identify potential clients but missing some revenue opportunities. More significantly, the number of mistakenly identified non-potential clients increased by 32.29%, which indicates a reduction in precision. However, there was an increase of 2.66% in the number of correctly identified non-potential clients, enhancing the amount of money saved by correctly classifying non-potential clients. The number of missed potential clients decreased by 17.66%, indicating improved sensitivity. This shift, while resulting in fewer identified potential clients, favored the identification of more non-potential clients, thereby improving overall effectiveness in avoiding unnecessary expenditures on unsuitable targets. Based on this performance analysis, the

classification reports in Figure 29 and Figure 30 provide additional information of the effectiveness of the SML model before and after its optimization, respectively.

Classification report for test dataset:

	precision	recall	f1-score
0	0.95	0.92	0.94
1	0.80	0.87	0.83
accuracy			0.91
macro avg	0.88	0.90	0.88
weighted avg	0.91	0.91	0.91

Figure 29. Classification Report Comparison Before Optimization.

Classification report for test dataset:

	precision	recall	f1-score
0	0.96	0.90	0.93
1	0.75	0.89	0.82
accuracy			0.90
macro avg	0.86	0.89	0.87
weighted avg	0.91	0.90	0.90

Figure 30. Classification Report Comparison After Optimization.

Initially, in Figure 29 the model showed high precision in identifying class 0 (potential clients) with a score of 0.95, and a lower precision of 0.80 for class 1 (non-potential clients), indicating a significant opportunity for improvement in minimizing unnecessary outreach to unsuitable leads. The recall of 0.92 for potential clients highlighted the model's effectiveness in capturing a high percentage of actual potential clients, although the number for non-potential clients decreased to 0.87.

Following the optimization, in Figure 30, there was an increase in precision for identifying potential clients, rising to 0.96. This improvement allows for a more effective and efficient allocation of resources toward those most likely to convert into clients,

ultimately enhancing the performance of the insurance company. At the same time, precision for identifying non-potential clients decreased from 0.80 to 0.75, suggesting an increase in misdirected efforts toward individuals less likely to convert. However, the recall for non-potential clients increased to 0.89, reflecting an enhanced ability of the model to identify genuine non-opportunities. This ensures that fewer non-potential clients are mistakenly pursued. These changes suggest a strategic shift in the model's focus, aiming to identify a higher number of non-potential clients at the expense of some accuracy in identifying actual potential clients.

4.4.2. Analysis of Financial Metrics

To illustrate the financial impact of implementing and optimizing the ML, the earning's percentage changes before and after the misclassification cost optimization are calculated in Table 9 in comparison to the original earnings of the insurance company.

Table 9. Percentage Changes from Before Implementing Any SML Model.

Stage	Earnings % Change
Before Misclassification Cost Optimization	67.93%
After Misclassification Cost Optimization	66.10%

Moreover, reflecting on the potential to adjust the threshold and its financial impact on the insurance company, it's important to recognize that the current setting is primarily based on expert judgment rather than statistical confidence intervals for earnings. Therefore, looking ahead, any adjustments to this threshold should be methodically tested within a controlled A/B testing framework to empirically determine the impact on earnings. Additionally, external factors such as economic inflation should be considered, as they could influence the generalizability and relevance of the results over time. Adopting this approach would provide a more rigorous validation of the threshold's efficacy and its real-world applicability under various economic conditions.

The necessity of fine-tuning the threshold settings, as discussed earlier, is more relevant when considering the financial results in the comparative analysis of the cost matrix

before and after misclassification cost optimization. Initially, the costs associated with TP (classifying potential clients correctly as Class 0) decreased about 1.09%. This reduction in expenses reflects a more efficient identification of genuine potential clients, likely aimed at enhancing long-term revenue through improved client acquisition and retention strategies. At the same time, costs related to FP (incorrectly classifying non-potential clients as Class 0) decreased about 14.39%. This decrease suggests a reduction in wasteful spending on client engagement efforts that are unlikely to generate incomes. Furthermore, the analysis reveals changes in handling FN, where potential clients were previously misclassified as Class 1 (non-potential clients). The costs in this category increased about 33.92%. This increase indicates a challenge in capturing potential clients, potentially leading to a loss of potential revenue. Additionally, the costs for TN (correctly identifying non-potential clients) also increased approximately 1.42%. This suggests a slight escalation in costs associated with managing non-potential clients, which might reflect a need for optimizing resource allocation and minimizing unnecessary outreach.

These shifts in the cost matrix post-optimization suggest a more precise and economically prudent approach in client classification. The increased investment in accurately identifying potential clients, despite the associated higher costs, aligns with a strategic focus on maximizing long-term value rather than merely reducing immediate expenses. However, the rise in costs associated with FP calls for further refinement of the model to ensure resources are judiciously used only on high-potential leads.

Before the misclassification cost optimization, the Predicted Potential Client (TP for potential clients and FP for non-potential clients) slightly decreased in correctly identifying potential clients and an increase in erroneously labeling non-potential clients as potential. In a similar way, in Predicted Non-Potential Clients (FP and TN), the numbers showed a significant reduction in FN, demonstrating the model's improved ability to correctly identify potential clients, while the increase in true TN indicates better accuracy in correctly identifying non-potential clients.

In conclusion, the refinement of the SML model has led to mixed results. While there has been a significant improvement in reducing false negatives and a slight increase in true negatives, the rise in false positives suggest that the model is becoming more sensitive in detecting potential clients but at the cost of more frequent false alarms.

5. Conclusions and Future Work

This master's thesis has explored the impact of applying a SML model for client segmentation within the automobile insurance sector. Using historical client data from said company, the classification report of the newly developed SML model, shows its precision and effectiveness: achieving an accuracy of 90%, with a precision of 96% for potential clients and 75% for non-potential ones, demonstrating a high reliability in the results of categorization of clients.

The SML model developed also demonstrated an important increase in the earnings of the insurance company after its full implementation, partly by decreasing the costs associated with misclassification of clients. The model's optimization of the misclassification cost threshold not only improved the model's accuracy but also increased its efficiency in terms of resource allocation, based on the 14% decrease in direct costs of classifying non-potential clients as potential, and an increase of almost 2% in avoided costs by accurately identifying non-potential clients.

However, a critical area for future research is the observed increase in FP costs in the model development, which translates into more potential clients being lost by being classified as non-potential. This is because obtaining information of new clients is difficult and challenging, which complicates the construction of a more precise model. Therefore, increasing the size of the database with additional variables and a higher number of new clients' data, would improve the results of the proposed SML model. Despite these challenges, the model performs well and has a positive financial impact on the insurance company. By effectively segmenting and classifying new clients and more accurately identifying non-potential clients, the model helps to save considerable amounts of money and resources.

References

Abolmakarem, S., Abdi, F., & Khalili-Damghani, K. (2016). Insurance customer segmentation using clustering approach. *International Journal of Knowledge Engineering and Data Mining, 1*(4), 18-39.

Avanijaa, J. (2021). Prediction of house price using xgboost regression algorithm. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(2), 2151-2155.

Bohnert, A., Fritzsche, A., & Gregor, S. (2019). Digital agendas in the insurance industry: the importance of comprehensive approaches. *The Geneva Papers on Risk and Insurance-Issues and Practice*(44), 1-19.

Boodhun, N., & Jayabalan, M. (2018). Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems, 4*(2), 145-154.

Castro, E. (2010, Julio-Diciembre). El estudio de casos como metodología de investigación y su importancia en la dirección y administración de empresas. *Revista Nacional de Administración, 2*(1), 31-54.

Chern, C. C., Lee, A. J., & Wei, C. P. (2015). Introduction to the special issue on "Data analytics for marketing intelligence. *Information Systems and e-Business Management*(13), 399-402.

Czajkowski, M., & Kretowski, M. (2016). The role of decision tree representation in regression problems–An evolutionary perspective. *Applied soft computing*(48), 458-475.

Dai, Y., & Wang, T. (2021). Prediction of customer engagement behaviour response to marketing posts based on machine learning. *Connection Science, 33*(4), 891-910.

Debener, J., Heinke, V., & Kriebel, J. (2023). Detecting insurance fraud using supervised and unsupervised machine learning. *Journal of Risk and Insurance, 90*(3), 743-768.

Eckert, C., Neunsinger, C., & Osterrieder, K. (2022). Managing customer satisfaction: digital applications for insurance companies. *The Geneva Papers on Risk and Insurance-Issues and Practice, 3*(47), 569-602.

Eluwole, O. T., & Akande, S. (2022). Artificial Intelligence in Finance: Possibilities and Threats. *EEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology* (pp. 268-273). IEEE.

Enholm, I. M., Papagiannidis, E., Mikalef, P., & Krogstie, J. (2022). Artificial intelligence and business value: A literature review. *Information Systems Frontiers, 5*(24), 1709-1734.

Galetsi, P., Katsaliaki, K., & Kumar, S. (2020). Big data analytics in health sector: Theoretical framework, techniques and prospects. *International Journal of Information Management*(50), 206-126.

Garg, M. C., & Garg, S. (2020). Operating Efficiency and Investment Efficiency: General Insurance Companies. *SCMS Journal of Indian Management*(131).

Guarda, T., Santos, M. F., Pinto, F., Silva, C., & Lourenço, J. (2012). A conceptual framework for marketing intelligence. *Journal of e-Education, e-Business, e-Management and e-Learning, 6*(2), 455.

Gupta, S., Ghardallou, W., Pandey, D. K., & Sahu, G. P. (2022). Artificial intelligence adoption in the insurance industry: Evidence using the technology–organization–environment framework. *Research in International Business and Finance*(63).

Haleem, A., Javaid, M., Qadri, M. A., Singh, R. P., & Suman, R. (2022). Artificial intelligence (AI) applications for marketing: A literature-based study. *International Journal of Intelligent Networks, 3*, 119-132.

Hallur, G. G., Prabhu, S., & Aslekar, A. (2021). Entertainment in era of AI, big data & IoT. *Digital Entertainment: The Next Evolution in Service Sector*, 87-109.

Hanafy, M., & Ming, R. (2021). Machine Learning Approaches for Auto Insurance Big Data. *Risks, 9*(2), 42.

Hedin, H., Hirvensalo, I., & Vaarnas, M. (2014). *The handbook of market intelligence: understand, compete and grow in global markets.*

Heeb, O., Barua, A., Menon, C., & Jiang, X. (2022). Building effective machine learning models for ankle joint power estimation during walking using FMG Sensors. *Frontiers in Neurorobotics, 16*.

Hill, L., Levy, F., Kundra, V., Laki, B., & Smith, J. (2015). *Data-driven innovation for growth and well-being.*

Hussain, K. &. (2016). Big data in the finance and insurance sectors. . *New horizons for a data-driven economy: A roadmap for usage and exploitation of big data in Europe*, 209-223.

Jones, K. I., & Sah, S. (2023). The Implementation of Machine Learning In The Insurance Industry With Big Data Analytics. *International Journal of Data Informatics and Intelligent Computing, 2*(2), 21-38.

Kasem, M. S., Hamada, M., & Taj-Eddin, I. (2024). Customer profiling, segmentation, and sales prediction using AI in direct marketing. *Neural Computing and Applications, 36*(9), 4995-5005.

Khodabandehlou, S., & Zivari, R. M. (2017). Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior. *Journal of Systems and Information Technology, 19*(1/2), 65-93.

Kotras, B. (2020). Mass personalization: Predictive marketing algorithms and the reshaping of consumer knowledge. *Big data & society, 7*(2).

Kumar, P., Taneja, S., & Mukul, &. Ö. (2023). Digital Transformation of the Insurance Industry–A Case of the Indian Insurance Sector. *The Impact of Climate Change and Sustainability Standards on the Insurance Market*, 85-106.

Lies, J. (2019). Marketing Intelligence and Big Data: Digital Marketing Techniques on their Way to Becoming Social Engineering Techniques in Marketing. *International Journal of Interactive Multimedia and Artificial Intelligence,, 5*(5).

Luciano, E., Cattaneo, M., & Kenett, R. (2023). Adversarial AI in Insurance: Pervasiveness and Resilience. *arXiv*.

Miklosik, A., & Evans, N. (2020). Impact of big data and machine learning on digital transformation in marketing: A literature review. *Ieee Access, 8*, 101284-101292.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis.*

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons, 4*, 51-62.

Rajagopal, N. K., Qureshi, N. I., Durga, S., Ramirez Asis, E. H., Huerta Soto, R. M., Gupta, S. K., & Deepak, S. (2022). Future of business culture: an artificial intelligence-driven digital framework for organization decision-making process. *Complexity*, 1-14.

Rana, A., Bansal, R., & Gupta, M. (2022). Big Data: A Disruptive Innovation in the Insurance Sector. *In Big Data Analytics in the Insurance Market* , 165-183.

Ranstam, J., & Cook, J. A. (2018). LASSO regression. *Journal of British Surgery, 105*(10), 1348-1348.

Ray, S., Thakur, V., & Bandyopadhyay, K. (2020). *India's insurance sector: Challenges and opportunities.*

Sahai, R., Al-Ataby, A., Assi, S., Jayabalan, M., Liatsis, P., & Loy, C. K. (2022). Insurance Risk Prediction Using Machine Learning. *The International Conference on Data Science and Emerging Technologies*, (pp. 419-433).

Saleh, A. M., A. M., & Kibria, B. G. (2019). *heory of ridge regression estimation with applications.*

Scardovi, C. (2017). Transformation in Insurance. *Digital Transformation in Financial Services*, 163-185.

Shah, D., & Murthi, B. P. (2021). Marketing in a data-driven digital world: Implications for the role and scope of marketing. *Journal of Business Research*(125), 772-779.

Shah, D., & Shay, E. (2019). How and why artificial intelligence, mixed reality and blockchain technologies will change marketing we know today. *Handbook of Advances in Marketing in an Era of Disruptions*, 377-390.

Shulla, K., & Leal-Filho, W. (2023). *Achieving the UN Agenda 2030: Overall actions for the successful implementation of the Sustainable Development Goals before and after the 2030 deadline.*

Shyam, R., & Chakraborty, R. (2021). Machine learning and its dominant paradigms. *Journal of Advancements in Robotics, 8*(2), 1-10.

Singh, J., & Singla, V. (2015). Big data: tools and technologies in big data. *International Journal of Computer Applications, 15*(112).

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of business research*(70), 263-286.

Sun, Q., Zhou, W. X., & Fan, J. (2020). Adaptive huber regression. *Journal of the American Statistical Association, 115*(529), 254-265.

Thontirawong, P., & Chinchanachokchai, S. (2021). Teaching artificial intelligence and machine learning in marketing. *Marketing Education Review, 31*(2), 58-63.

Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N., & Campos-Cordobés, S. (2018). Big Data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems, 8*(12), 742-755.

Turkmen, B. (2022). Customer Segmentation with machine learning for online retail industry. *The European Journal of Social & Behavioural Sciences*.

United Nations. (2024). *Transforming our world: the 2030 Agenda for Sustainable Development*. Sustainable Development: https://sdgs.un.org/2030agenda

Vassakis, K., Petrakis, E., & Kopanakis, I. (2018). Big data analytics: applications, prospects and challenges. *Mobile big data: A roadmap from models to technologies*, 3-20.

Yildirim, E., Cam, V., Balki, F., & Sarp, S. (2022). Sales Forecasting During the COVID-19 Pandemic for Stock Management. *Machine Intelligence and Digital Interaction Conference*, (pp. 111-123).

Yum, K., Yoo, B., & Lee, J. (2022). Application of AI-based customer segmentation in the insurance industry. *Asia Pacific Journal of Information Systems, 32*(3), 496-513.

Zhang, S., Liao, P., Ye, H. Q., & Zhou, Z. (2022). Dynamic marketing resource allocation with two-stage decisions. *Journal of Theoretical and Applied Electronic Commerce Research, 17*(1), 327-344.

Zhou, K. F., & Yang, S. (2016). Big data driven smart energy management: From big data to big insights. *Renewable and sustainable energy reviews*(56), 215-225.