CUSTOMER CHURN PREDICTION FOR A PERSONAL CARE PRODUCT RETAIL CHAIN OPERATING IN TURKEY

ERCAN IŞIK

MEF UNIVERSITY

SEPTEMBER 2024

MEF UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
MASTER'S IN INFORMATION TECHNOLOGIES

M. Sc. THESIS

CUSTOMER CHURN PREDICTION FOR A PERSONAL CARE PRODUCT RETAIL CHAIN OPERATING IN TURKEY

Ercan IŞIK

ORCID NO: 0000-0003-0916-6309

Thesis Advisor: Asst. Prof. Dr. Tuna ÇAKAR

SEPTEMBER 2024

ACADEMIC HONESTY PLEDGE

I declare that all the information in this study is collected and presented in accordance with academic rules and ethical principles, and that all information and documents that are not original in the study are referenced in accordance with the citation standards, within the framework required by the rules and principles as a graduation project Master's Degree in Information Technologies.

Name and Surname: Ercan IŞIK

Signature:

ABSTRACT

CUSTOMER CHURN PREDICTION FOR A PERSONAL CARE PRODUCT RETAIL CHAIN OPERATING IN TURKEY

Ercan IŞIK

M.Sc. in Information Technologies

Thesis Advisor: Asst. Prof. Dr. Tuna ÇAKAR

September 2024, 42 Pages

Understanding the reasons for customer loss and the customer behaviors leading to it, as well as being able to predict customer's loyalty to an industry or a company provides enormous advantages in retaining existing customers and avoiding revenue loss due to the marketing and advertising costs associated with attracting new customers. In this study, the 29-month data from a personal care product retail chain operating in Turkey was used, and because of the imbalanced values and non-customer entries of the dataset, the oversampling method and synthetic sampling was applied.

During the model development phase, Logistic Regression, Decision Tree, K-Nearest Neighbors, Random Forest, Extra Trees Classifier, and MLP (Multi-Layer Perceptron) Classifier were applied, and their performances were evaluated using metrics such as accuracy, recall, F1-score, precision, and confusion matrix. Based on these comparisons, it was observed that the Random Forest and MLP Classifier models demonstrated the best performances for this dataset, while other tree-based algorithms, such as the Extra Trees Classifier and Decision Tree, achieved slightly lower but comparable performance.

Keywords: Customer churn, churn prediction, machine learning, customer retention

Numeric Code of the Field: 92404

ÖZET

TÜRKİYE'DE FAALİYET GÖSTEREN BİR KİŞİSEL BAKIM ÜRÜNLERİ MARKET ZİNCİRİ İÇİN MÜŞTERİ KAYBI TAHMİNİ

Ercan IŞIK

Bilişim Teknolojileri Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Tuna Tuna ÇAKAR

Eylül 2024, 42 Sayfa

Müşteri kaybının nedenlerini ve buna yol açan müşteri davranışlarını anlamak, ayrıca müşterinin bir sektöre veya şirkete olan sadakatini tahmin edebilmek, mevcut müşterileri elde tutmada ve yeni müşterilere ulaşmak için yapılan pazarlama ve reklam maliyetlerinden kaynaklanan gelir kaybını önlemede büyük avantaj sağlar. Bu çalışmada, Türkiye'de faaliyet gösteren bir kişisel bakım ürünleri perakende zincirine ait 29 aylık veri kullanılmış; veri setindeki dengesiz dağılım ve müşteri olmayan girişler nedeniyle aşırı örnekleme ve sentetik örnekleme yöntemleri uygulanmıştır.

Model geliştirme aşamasında Lojistik Regresyon, Karar Ağacı, K-En Yakın Komşu, Rassal Orman, Ekstra Ağaç Sınıflandırıcı, MLP (Çok Katmanlı Algılayıcı) Sınıflandırıcı uygulanmış ve doğruluk, geri çağırma, F1 skoru, kesinlik ve karmaşıklık matrisi gibi metrikler kullanılarak performansları değerlendirilmiştir. Bu karşılaştırmalar sonucunda, Rassal Orman ve MLP Sınıflandırıcı modellerinin bu veri seti için en iyi performansı gösterdiği gözlemlenmiş; Ekstra Ağaç Sınıflandırıcı ve Karar Ağacı gibi diğer ağaç tabanlı algoritmaların ise biraz daha düşük fakat karşılaştırılabilir performans sağladığı tespit edilmiştir.

Anahtar Kelimeler: Müşteri kaybı, kayıp tahmini, makine öğrenimi, müşteri tutma.

Bilim Dalı Sayısal Kodu : 92404

Tüm süreç boyunca desteğini esirgemeyen

eşim Şenay DEMİRHAN IŞIK`a ve aileme ithaf ediyorum.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis professor, Asst. Prof. Tuna Çakar, who contributed greatly to the completion of this study with his valuable guidance and knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

NN : Neural Networks

LLM : Logit Leaf Model

KNCn : K-neighbors with the variants of centroids

KNCa : K-neighbors with principal component analysis

DT : Decision Trees

OCP : One Class Predictor

GBM : DT with boosting based in gradients Gradient Boosting Machines

LR : Logistic Regression

LRVC : Logistic Regression with cross validation

SVM : Support Vector Machine

SVM rbf : SVM with a radial basis function kernel

RF : Random Forest

KNN : K Nearest Neighbors

CRM : Customer Relationship Management

CHAID : Chi-square Automatic Interaction Detector

ML : Machine Learning

MLP : Multi-layer Perceptron

EDA : Exploratory Data Analysis

MAE : Mean Absolute Error

MSE : Mean Squared Error

RMSE : Root Mean Squared Error

R^2 : R-Squared

TP : True Positive

FP : False Positive

FN : False Negative

TN : True Negative

INTRODUCTION

Problem to be Solved

The rapid growth of digital platforms and the emergence of numerous alternatives to traditional retail services have intensified competition within the e-commerce industry. In today's highly competitive market, retaining customers, minimizing the cost associated with customer churn, and targeting the right customer segments are critical challenges that many companies are seeking to address. This is particularly relevant for companies like the one investigated in our study, where understanding and predicting customer churn is essential to maintaining a competitive edge.

The Aim of the Thesis

This thesis aims to uncover customer value and predict potential customer churn through the analysis of customers' transaction behavior using data science and data analytics methods.

The Importance of the Thesis

By identifying the customers most likely to cease transactions with the company allows businesses to implement targeted measures to retain and tailor their marketing strategies based on these predictions.

Limitations of the Study

A nationwide operating company provides 29-months of imbalanced data, which includes non-customer entries such as cashiers from physical stores, that are not distinguished within the dataset.

Thesis Outline

This section provides a summary of the thesis, highlighting the problem addressed, the purpose and significance of the study, as well as its limitations. The remaining sections are outlined below.

Literature Review explores the existing definitions of customer churn found in the literature and the strategic benefits it offers to businesses. It also examines the machine learning techniques previously used in this research to predict customer churn, highlighting their effectiveness in similar contexts.

Methodology provides a theoretical overview of the approaches used for predicting customer churn. It shows the various stages involved in churn prediction modeling and details the specific machine learning techniques utilized in this study.

Model Development and Validation provides examining and understanding the dataset used in this research, preparing the dataset to apply the machine learning methods and evaluating and comparing the models performance and accuracy.

Discussion and Conclusion sections present a comprehensive analysis of the study's findings and the key insights drawn from the research. It concludes with the final interpretations of the results, along with recommendations for future research and practical applications.

LITERATURE REVIEW

Definition of Churn Rate

The churn rate is a key metric that reflects customer retention and overall business health. It indicates how well a company maintains its customer base and can reveal underlying issues such as poor customer service, inadequate product quality, or competitive pressures. The churn rate is calculated as a percentage, with the formula

$$\left(\frac{Number\ of\ Customers\ Lost\ During\ Period}{Number\ of\ Customers\ at\ Start\ of\ Period}\right) \times 100$$

The churn rate helps businesses quantify customer loss and monitor trends over time.

Simply, customer churn means the customer is lost to the company, and churn rate shows how much customer did churn during the period.

Significance of Customer Churn

The churn rate is an important metric that directly affects revenue, customer lifetime value (CLV), and overall business stability. A high churn rate leads to revenue loss and increased costs for acquiring new customers, while a low churn rate enhances revenue stability and increases CLV, making a business more financially sustainable and attractive to investors. Managing churn rate is also useful in predicting future revenue streams, improving customer satisfaction, and better competitive strategies. Lower churn rates lead to strong customer relationships, contributing to loyalty, repeat purchases, and long-term business success. Understanding and managing churn effectively ensures stable revenues, higher customer value, and sustainable growth in a competitive market [1].

Factors That Causes Customers to Churn

Various factors can be the reason for high churn rates, including weak customer service, bad product quality, and competitive pressures. Issues like slow or non-effective

customer support, products that can’t meet expectations, and better offerings from competitors lead customers to leave. Pricing problems, such as overpricing or aggressive competitor discounts, can also make customers churn. External factors like economic conditions and shifting customer preferences further influence churn. To reduce churn, businesses must adopt a comprehensive strategy that enhances customer satisfaction, maintains competitive offerings, optimizes pricing, fosters engagement, and adapts to market changes.

Related Works

Azeem, Usman, and Fong developed an improved PCA-AdaBoost model to address challenges in predicting customer churn within the e-commerce industry, where customers are typically non-contractual. They incorporated two additional features—transaction length and customer satisfaction—into the RFM (Recency, Frequency, Monetary) analysis to enhance its effectiveness in predicting churn. They demonstrated that their PCA-AdaBoost model outperforms traditional algorithms, such as logistic regression, SVM, and standard AdaBoost, in terms of accuracy and stability when dealing with high-dimensional and imbalanced datasets. However, they also identified several limitations in their approach. They noted that the RFMLS framework is not the only effective method for predicting churn, and that their model's computation time is longer than that of simpler models due to its integration of multiple decision-tree classifiers [2].

Xiaojun Wu and Sufang Meng addressed the imbalance in e-commerce customer churn data, where churn samples typically far exceed non-churn samples, causing classifiers to be biased towards the majority class. Recognizing that retaining existing customers is much more cost-effective than acquiring new ones, they proposed an improved SMOTE technique to balance the dataset by generating a controlled number of positive and negative samples. This technique involved setting a sampling ratio to manage model training time effectively. They then applied the balanced dataset to the AdaBoost algorithm to train a weak classifier, which iteratively improved the prediction accuracy for both churn and non-churn customers. Their empirical study on a B2C e-commerce platform demonstrated that their model achieved better efficiency and accuracy compared

to existing customer churn prediction algorithms. They concluded that the improved SMOTE method provides a valuable reference for e-commerce customer churn prediction and could be applicable in other fields as well [3].

P. Nagaraj, V. Muneeswaran, A. Dharanidharan, M. Aakash, K. Balananthanan and C. Rajkumar constructed a customer churn prediction model for e-commerce using data collected from the Kaggle platform, which included customer consumption information. They began by preprocessing the data to remove any null values, ensuring optimal performance for their models. The data was then split into training (75%) and testing (25%) sets.

They applied three different machine learning algorithms—Support Vector Machine (SVM), Decision Tree, and Random Forest—to predict customer churn. The Random Forest algorithm achieved the highest accuracy among these. They demonstrated that using machine learning techniques like these can significantly improve customer retention by accurately predicting churn, allowing businesses to address issues proactively [4].

Ishrat Jahan and Dr. Tahsina Farah Sanam developed a customer churn prediction framework for e-commerce using various classifiers to identify the best model for predicting churn with high accuracy. Their framework included five key components: exploratory data analysis (EDA), data preprocessing, model tuning, comparison of different models, and generating insights and recommendations. They found that the CatBoost classifier achieved the highest performance, with 100% accuracy and 100% F1-score. After selecting CatBoost as the best model, they applied recursive feature elimination (RFE) to rank the importance of features, providing valuable insights. Their study contributes to the existing literature by presenting a comprehensive approach to customer churn prediction in e-commerce, addressing data imbalance and missing values through preprocessing and optimizing model performance with hyperparameter tuning. They suggest that future work could explore more advanced machine learning algorithms and techniques to further enhance the system's performance [5].

1. METHODOLOGY

1.1 Machine Learning Methods

Alex Bekker from ScienceSoft states the importance of machine learning algorithms in order to get accurate churn prediction. He claims that several machine learning techniques can be efficiently utilized to forecast possible churn of customers by considering common customer behaviors and transactional patterns, hence signaling potential churners [6].

1.1.1 Logistic Regression

Regression is one of the statistical methods used to analyze and predict the relationships between variables. Logistic regression, a specific type of regression analysis, is employed to estimate the probability of an event occurring. While linear regression is useful for predicting continuous data, logistic regression serves as a probabilistic statistical model designed for classification tasks. It is particularly used for binary classification, where a categorical outcome depends on one or more variables. This algorithm is also applicable in scenarios such as predicting customer churn, which is a binary classification problem.

The predicted output of the logistic regression is in the expression below.

$$L_i = In\left(\frac{P_i}{1-P_i}\right) = Z_i = \beta X_i + u_i$$

In the expression above, the term $\frac{P_i}{1-P_i}$ represents the likelihood of an event happening (P_i) compared it to not happening $(1-P_i)$. Hence, $In\left(\frac{P_i}{1-P_i}\right)$ is the logit transformation which converts probabilities into a continuous range of values. Furthermore, βX_i represents the weighted sum of input variables (X_i) multiplied by their corresponding coefficients (β) and u_i represents the error term [7].

1.1.2 Decision Tree

Decision Tree is one of the many supervised machine learning techniques. While it is predominantly used for classification problems, it can also be applied to regression analysis. The method involves creating a tree-like structure by branching out from a root node. In this structure, internal nodes represent dataset features, branches correspond to decision rules, and leaf nodes signify outcomes [8].

1.1.3 KNN (K-Nearest Neighbors)

K-Nearest Neighbors (KNN) is a widely used classification technique in machine learning. It is a simple method that requires minimal input: a parameter k, a labeled training dataset, and a distance feature to calculate the similarity between data points in an n −dimensional space.

Mehmed Kantardzic states that, the classification process of KNN typically includes these steps [9]:

1. **Determine the parameter k**: The value of k, representing the number of nearest neighbors to consider, is selected. This parameter influences the model's sensitivity to the local structure of the data.
2. **Calculate Distances**: For each data point in the test set, the distance to every point in the training set is calculated. Common distance metrics include Euclidean, Manhattan, or Minkowski distances, depending on the nature of the data.
3. **Identify Nearest Neighbors**: Once distances are calculated, the data points in the training set are sorted by distance to the test point. The k closest neighbors are identified based on this sorted list.
4. **Determine Class of Neighbors**: The class or category of each of the k nearest neighbors is identified. Each neighbor contributes to the final classification decision.
5. **Majority Vote for Classification**: The test sample is classified based on a majority vote from its k nearest neighbors. The class that appears most frequently among these neighbors is assigned as the predicted class for the test sample.

This process allows KNN to classify new data points based on the categories of the nearest examples in the training data, making it particularly useful for tasks where the decision boundaries between classes are complex or irregular.

1.1.4 Random Forest

Random forests are a traditional classification method built on the concept of random subspaces. This approach involves constructing multiple decision trees and combining them using random selection of subspaces and bagging techniques, which enhances the overall model's performance and robustness [10, 11].

Leo Breiman describes random forests as a method that consists of a collection of tree-structured classifiers, denoted as $h(x, k)$, where $k = 1, \ldots$ with each tree being generated using independently and identically distributed random vectors. Each tree contributes one vote for the most frequent class when making a prediction for a given input x [12].

Random forests are particularly well-suited for binary classification tasks, such as predicting customers' transaction behaviors or determining customer loyalty. The ensemble of decision trees within the random forest structure helps improve accuracy and reduces the risk of overfitting compared to individual decision trees [11].

1.1.5 Extra Tree Classifier

A machine learning technique named The Extra Trees classifier, shares some minor and major points with Random Forests. In Extra Trees, the approach is to create decision trees with no bootstrapping, leading to a result that all trees are completed training on the whole dataset. In addition to that, Extra Trees differ from traditional decision trees in splitting nodes [13].

The Extra Trees classifier chooses splits at random, in contrast to standard decision trees, which choose the best split based on factors like information gain or Gini impurity. By adding more randomness to the tree-building process, this random splitting produces a more varied ensemble of trees. The primary goal of this added randomness is to reduce

the risk of overfitting and enhance the model's robustness, especially when working with noisy or high-dimensional datasets. By randomizing the splits, Extra Trees can achieve greater generalization and stability in its predictions.

1.1.6 AdaBoost

AdaBoost, or adaptive boosting, is an ensemble learning technique that builds a strong prediction model by combining several weak classifiers. It operates by repeatedly applying a classification algorithm, each time reweighting the training data, to various iterations of the data. Higher weights are assigned to instances that prior classifiers misclassified, enabling the model to concentrate on the more challenging samples. A weighted majority vote or an average of the predictions from each classifier are used to determine the final prediction [3].

1.1.7 MLP Classifier

The MLP Classifier is a type of artificial neural network widely used for classification tasks, including customer churn prediction. It consists of multiple layers: an input layer, one or more hidden layers, and an output layer. The key advantage of MLP over simpler models like Logistic Regression is its ability to capture complex, non-linear relationships between features and the target variable.

MLP Classifier is trained using backpropagation, where the model iteratively adjusts its parameters to minimize prediction errors. Non-linear activation functions such as ReLU or Sigmoid help the model learn from intricate patterns in the data. This makes it well-suited for tasks like churn prediction, where customer behavior often involves hidden complexities.

Studies have demonstrated the effectiveness of neural networks in churn prediction, particularly when traditional models fall short [17].

1.2 Exploratory Data Analysis

In data-driven analytics, exploratory data analysis, or EDA, is a crucial phase where visualization techniques are applied for understanding and summarizing the data.

EDA aims to uncover hidden patterns, relationships, and insights that can guide the ensuing phases of supervised and unsupervised machine learning modeling by visualizing and examining the relationships between the dataset's features [9, 10].

In EDA, visualization is an essential tool that helps analysts and researchers comprehend the dataset in great detail. Plots such as scatter plots, histograms, box plots, and heat maps can be used to visualize complex relationships between variables. Finding trends, understanding variable distributions, identifying outliers, and possibly spotting patterns that could direct the modeling process are all made easier by visualizing the data.

In order to lay a strong basis for subsequent modeling stages, EDA is essential. We can make well-informed decisions about feature engineering, data transformations, and model selection by gaining insights into the properties, distributions, and relationships between variables in the data. This preliminary analysis offers crucial information for developing reliable and accurate customer churn prediction models.

1.3 Feature Engineering

The effectiveness of machine learning models is greatly impacted by feature engineering, a crucial data processing method. In order to better suit the requirements and features of the selected machine learning algorithm, it involves transforming or adding new features from the original dataset. When feature engineering is done right, it enhances algorithm performance and helps extract meaningful information from the data [14].

For the company`s churn prediction purposes, feature engineering is the most important and the major step in preparing the dataset for modeling. To maximize the prediction accuracy of our model, we carefully selected raw features, transformed them into more efficient versions according to the observations of EDA, and created new features by means of existing features. By these accomplishments, we uncovered specific and significant customer behavior patterns and relationships related to customer’s churn to provide better prepared information for our machine learning model.

To summarize all, feature engineering is an important phase in data processing that affects the machine learning algorithms’ performance highly. By transforming, creating, and

selecting features, we revised the dataset to better fit with the requirements of the problem we aim to solve, and uncover valuable insights. In order to predict possible customer loss for the company in our study, feature engineering stands as one of the leading factors to accomplish higher accuracy results.

1.4 Performance Metrics

The performance of the models is evaluated with metrics such as accuracy, precision, recall, and F1 score. Beside these metrics, we also used confusion matrices to visualize the accuracy and the performance of our models. These metrics are very important to observe, understand and improve the classification performance of the models.

The confusion matrix provides a concise summary of a machine learning classification model's performance by comparing the predicted values with the actual values [7].

Confusion Matrix		**Actual Values**	
		0	**1**
Predicted Values	**0**	True Positive (TP)	False Positive (FP)
	1	False Negative (FN)	True Negative (TN)

Table 1.1:The confusion matrix

TP (True Positive): Represents cases where a sample with an actual class value of 1 is correctly predicted as 1.

TN (True Negative): Refers to situations where a sample with an actual class value of 0 is correctly predicted as 0.

FN (False Negative): Occurs when a sample with an actual class value of 1 is incorrectly predicted as 0.

FP (False Positive): Happens when a sample with an actual class value of 0 is incorrectly predicted as 1 [3, 7].

The metrics we used to analyze the performance of our models is calculated as the following:

Accuracy refers to the ratio of correctly predicted samples to the total number of samples.

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

Precision is the proportion of correctly predicted positive samples to the total number of samples classified as positive.

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

Recall is essentially the ratio of true positives to the number of all positive samples.

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

F1 – score is calculated by means of harmonic mean of precision and recall values.

$$F1 - score = \frac{2.Recall.Precision}{Recall + Precision}$$

2. MODEL DEVELOPMENT AND VALIDATION

2.1 Dataset

The dataset includes roughly 2,5 years of transaction-based data of the customers from a nationwide operating company. It contains entries that each of them represents the goods that were purchased in a single receipt. In total, there are 59139246 entries and these entries represent 16030534 numbers of purchases made by 5623356 unique customers. The initial features of our dataset are as follows:

- **TRANS_ID** : Transaction ID
- **TRANS_DATE** : Date of the transaction
- **STORE_CODE** : Code of the store where the transaction occurred
- **CUST_ID** : Customer ID
- **PRODUCT_CODE** : Code of the product purchased
- **BARCODE** : Barcode of the product
- **AMOUNT** : Total amount of the transaction
- **UNIT_PRICE** : Price per unit of the product
- **NO_DISCOUNT** : Indicates whether a discount was applied or not
- **DISCOUNTED_UNIT_PRICE** : Unit price after discount
- **DISCOUNTED_TOTAL_PRICE** : Total price after discount
- **TOTAL_DISCOUNT_AMOUNT** : Total amount of discount applied
- **UNIT_CAMPAIGN_DISCOUNT** : Discount amount due to a campaign per unit

2.2 Exploratory Data Analysis of Available Data

In this first phase of the study, the relationship between categorical variables and churn were explored through. In order to highlight and discuss predominant factors that might lead to customer churn, following visualizations were employed. The conclusions drawn from the visualization played a key role in training the data set.

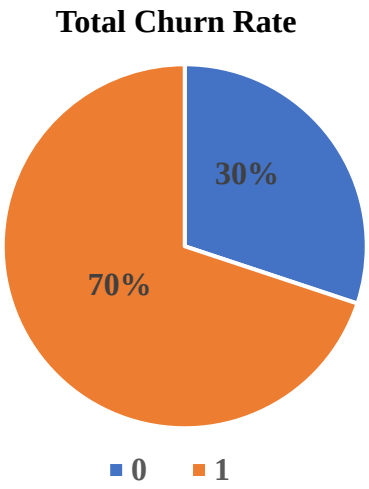

Figure 2.1: Customer's churn rate

From Figure 2.1, it is evident that 30.1% of customers remain active in our dataset, but the 69.9% is churned. It's clear that churn classes are imbalanced and this situation should be handled in further processing phases.

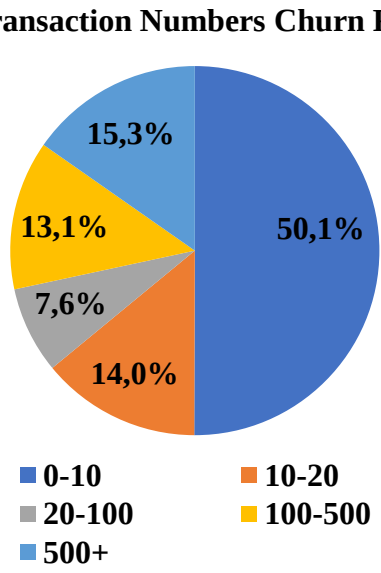

Figure 2.2: Customer's churn rate based on the transactions

When we look at the churn rate based on transactions of customers in Figure 2.2, it is clear that the churn rate decreases with more transactions until the transaction number hits roughly 100, then the churn rate begins to increase again. This inconsistency is probably due to the dataset entries of sales made by physical store employees to users who do not have a customer number.

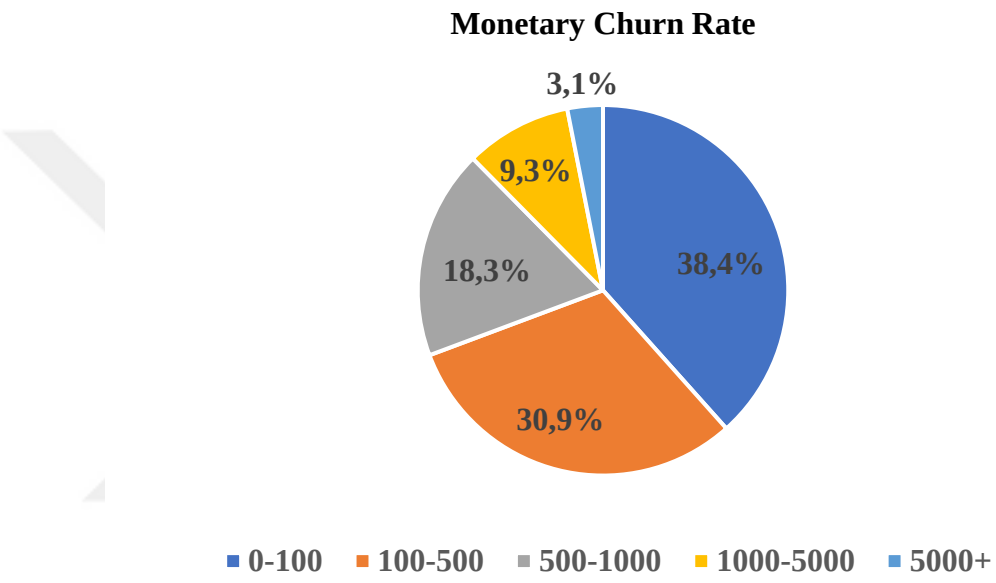

Figure 2.3: Customer's churn rate based on the monetary

From the Figure 2.3, it can be clearly seen that the churn rate decreases with more money spent, as expected. With this observation, we can implement some features based on the money spent by customers in the feature engineering phase to power up the training dataset. However, the entries of sales made by physical store employees for users affects these values and this situation should be considered while processing data. Since customer IDs of these employees are unavailable, these entries should be filtered out based on their specific values.

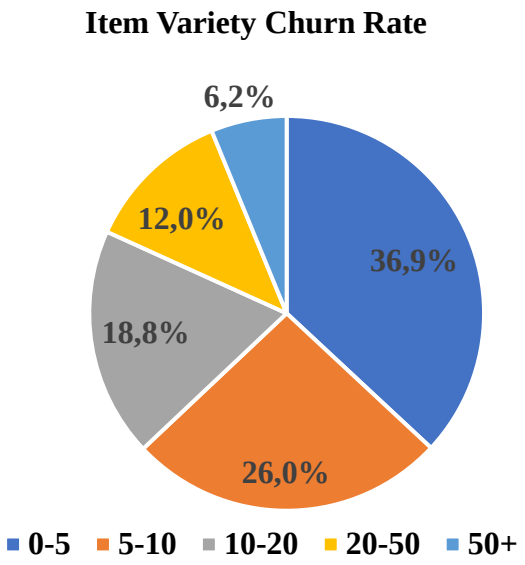

Figure 2.4: Customer's churn rate based on the item variety

From Figure 2.4 we observe that the churn rate decreases with more different items. This leads us to the conclusion that customers who purchase different products more frequently are less likely to churn.

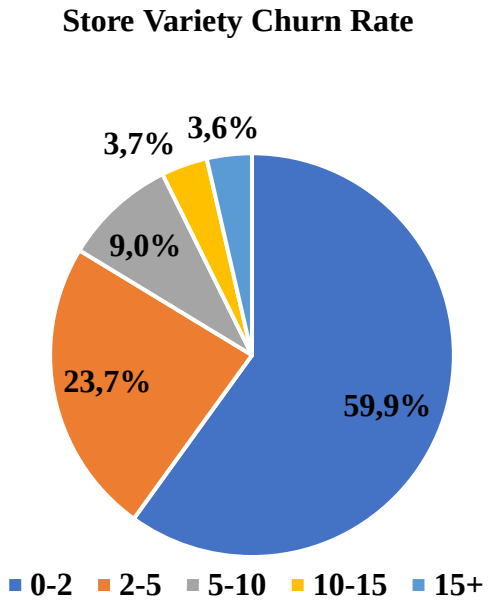

Figure 2.5: Customer's churn rate based on the store variety

When we look at the figure 2.5, which is the churn rate based on the store variety value of customers, the chart shows that the churn rate decreases with purchases from different stores, as expected. The observation states that features based on the store variety of customers in the feature engineering phase might power up the training dataset. The entries of sales made by physical store employees do not directly affect this feature since one employee is most likely to make sales in one store unless in specific situations.

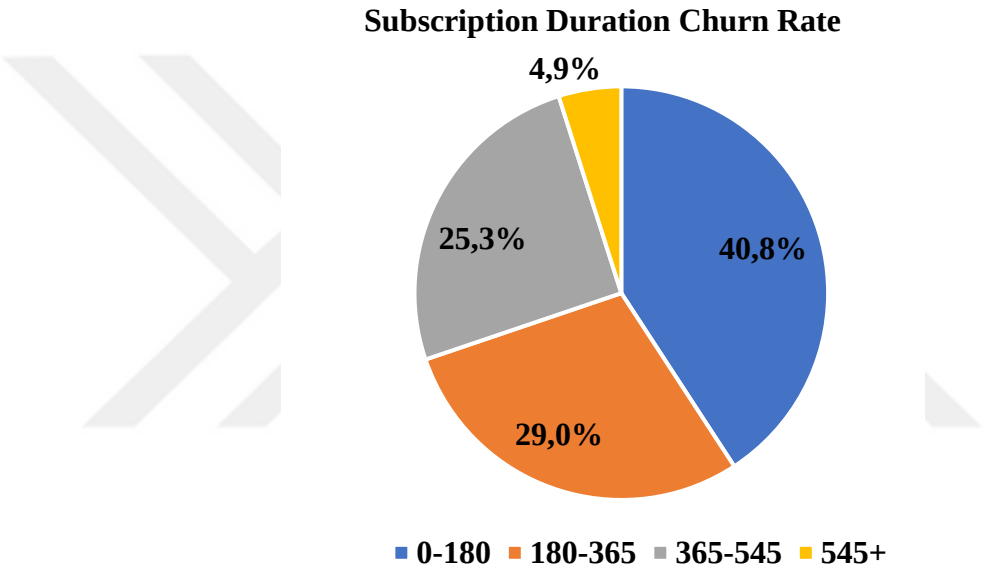

Figure 2.6: Customer's churn rate based on the subscription duration in days

Figure 2.6 clearly shows that the churn rate decreases with increasing subscription period. So, it can be said that long-time customers are less likely to churn. With this information, we can say that a subscription duration feature for the train dataset is likely to increase our model's predictive power.

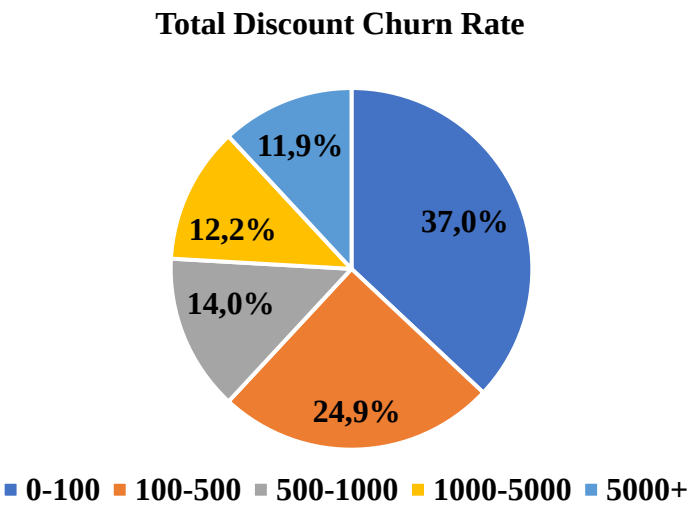

Figure 2.7: Customer's churn rate based on the total discount amount

Figure 2.7 illustrates a clear inverse relationship between the amount of discounts used by customers and their likelihood to churn. The data reveals that as customers take advantage of higher total discount amounts, their churn rates significantly decrease. Customers who receive greater savings through discounts appear to be more satisfied with their purchasing experience, leading to increased loyalty and a reduced likelihood of churn. The trend highlighted in this figure emphasizes the potential of discount strategies as an effective tool for minimizing churn and fostering long-term customer relationships.

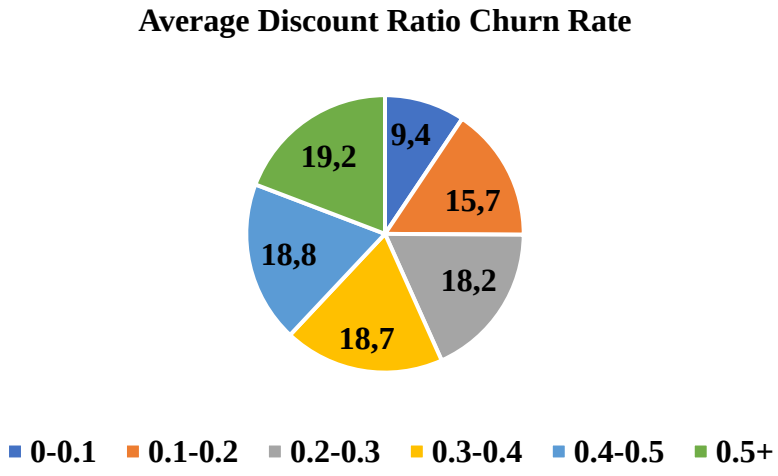

Figure 2.8: Customer's churn rate based on the average discount rate

Interestingly, while Figure 2.7 showed that higher total discount amounts are associated with lower churn, Figure 2.8 reveals the opposite trend when examining the average discount rate. As the average discount rate increases, customer churn tends to rise slightly. This phenomenon may be attributed to the presence of less loyal customers who engage in transactions primarily when significant discounts are offered. These customers may only make purchases during promotional events, indicating a transactional relationship rather than brand loyalty. Consequently, when discounts become less frequent or less substantial, these customers are more likely to churn. This contrast between total discount amounts and average discount rates underscores the complexity of customer behavior and suggests that discount strategies must be carefully considered to effectively influence retention.

2.3 Preprocess Phase

2.3.1 Approach to Problem

During the preparation of the dataset for model training, several steps were undertaken. Since the information on whether users had churned was not readily available in the dataset, the last purchase date up to six months before the dataset's final date was used as the reference date. Based on whether a user made a purchase after this reference date, it was determined whether the user had churned or not. In order to address the imbalance between churn and non-churn classes, synthetic sampling was performed for the minority class. This was achieved by gradually moving the reference date back by one month at a time, from the user's last purchase date to their first purchase date, and applying the same process repeatedly. By doing so, synthetic examples were generated for the minority class, helping to mitigate the class imbalance to some extent. Additionally, users' purchasing behaviors were analyzed on a monthly basis for the four months leading up to the reference date, and average values were examined for all purchases made from the reference date back to their first purchase date. All this information was stored in a new dataset to train the models.

2.3.2 Necessary Libraries for Preprocessing

In the preprocessing phase of our project, we utilized the “date-time” and “time” libraries to accurately handle dates, and the “DateOffset” class from the pandas.tseries.offsets module to enhance the precision of date filtering. For general data processing, we employed the “pandas” library, and for data visualization, we used the “matplotlib” library. Additionally, to assist with memory management, we incorporated the “gc” (garbage collector) built-in library.

2.3.3 Reading Raw Data

At this juncture, the dataset is initially ingested in segmented portions to manage its size efficiently. Upon completion of the data loading process, we proceed to address the numerical values where the decimal separator is represented by a comma, replacing it with a period to ensure accurate interpretation. This is followed by converting the data from string format to numeric types.

Subsequently, the "TRANS_DATE" column, which records transaction dates, is converted from a string to a datetime format to facilitate more effective temporal analysis. We then apply filters to remove entries with invalid customer identifiers and those with erroneous pricing data. Finally, variables that are no longer required are deleted, and the garbage collector is invoked to optimize memory management.

With these preparatory steps completed, the dataset is now ready for further analytical processing.

2.3.4 Filtering outliers

To effectively select the values we intend to filter, we analyzed customer’s all-time purchases.

Through this approach, we acquire data related to the total number of transactions each customer has made, the total amount they have spent, and the total variety of distinct products they have purchased. When we attempt to confine these values within specific ranges and visualize them, the results we encounter are as follows:

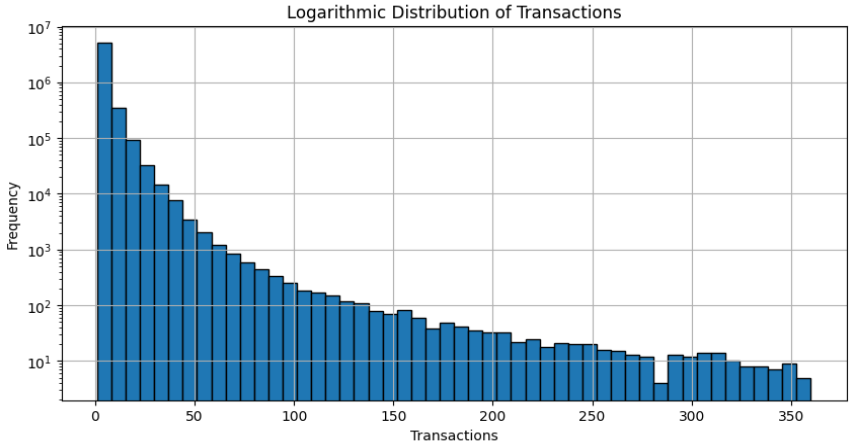

Figure 2.9: Distribution of customers based on transactions less than 360

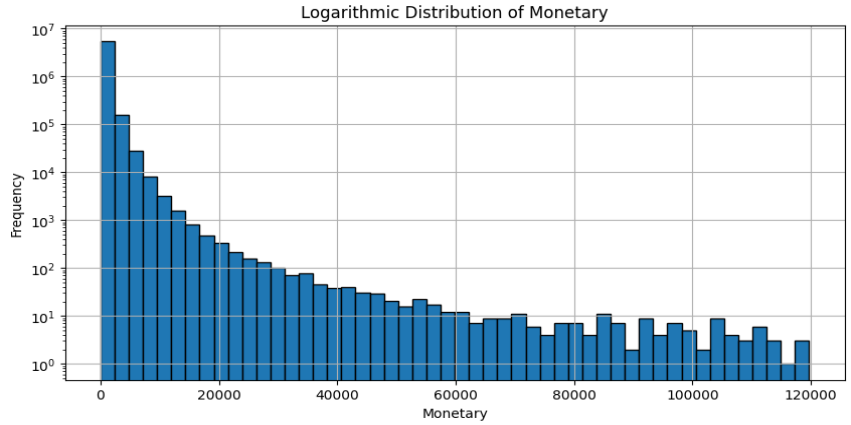

Figure 2.10: Distribution of customers based on monetary less than 120.000

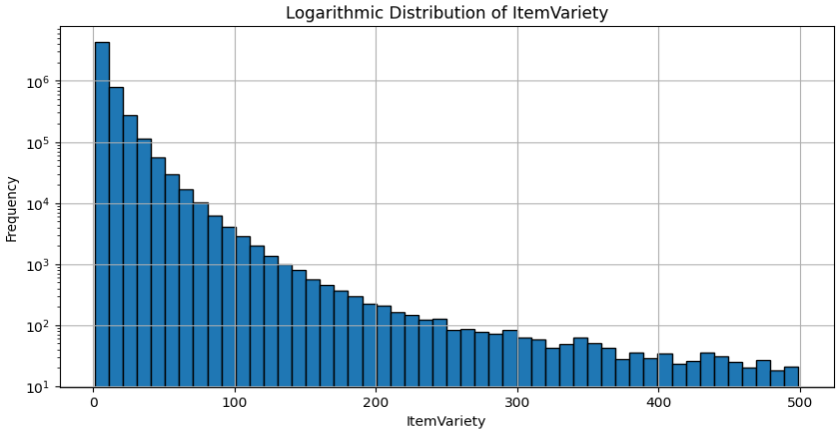

Figure 2.11: Distribution of customers based on ıtem variety less than 500

Based on these criteria, we filtered out users who had fewer than 10-100 transactions or whose values fell outside the range of what is considered significant compared to the dataset's time span, these values are:

- Less than 360 transactions,
- Less than 60000 liras spent
- Less than or equal to 500 different items purchased

Accordingly, we assume that over approximately 2.5 years, a noteworthy customer would have made at least 20 transactions and spent more than 1000 TL. Tightening the filters on the lower end of this range is expected to enhance model accuracy, but it may also make it more challenging to predict churn for customers who shop less frequently. According to observations from previous iterations, no such filtering was applied in the initial stages, and a model accuracy of around 0.73 was achieved, yet further improvement beyond this value was not possible.

2.3.5 First phase of preprocessing

2.3.5.1 Necessary Features before monthly-splitting

In the first phase of preprocessing, we split the entire dataset into months, recording and combining the necessary cumulative metrics from transactions made within and up to those months. Subsequently, we extracted the required data based on users' reference dates from this dataset, thereby simplifying our process. For instance, the total number of transactions made by users in January 2022 were stored in the "Frequency-2022-01-01" column of our dataset. In order to achieve that, we first converted the total payments made by users in their transactions into a numeric format using the "DISCOUNTED_TOTAL_PRICE" column, storing these values under the "Monetary" label.

Next, we recorded the dates of users' first and last orders. The last purchase date was used as the initial reference date in the second preprocessing stage, while the first purchase date helped us to calculate user's subscription duration up to that reference date

during the second preprocessing stage. Furthermore, in order to optimize memory management, we deleted the variables used to calculate these metrics and invoked the garbage collector.

Since we were not able to utilize data that did not allow us to look back four months from the reference date, we excluded users whose last purchase occurred after May 1, 2022, from the dataset.

Finally, we calculated metrics such as the discount rate per transaction and the total discount, preparing the dataset for the next stage, where it was split by months.

2.3.5.2 Monthly-splitting

Since our dataset covered a 29-month period, we initially obtained 29 separate data segments. We started with the first month of 2022, the beginning of our dataset, and selected the date ranges for filtering accordingly.

After determining the date ranges, we calculated the cumulative metrics up to that point, including "PrevTotalMonetary," "PrevTotalFrequency," and "PrevItemVariety." These metrics were then added to the filtered monthly data. The resulting data, along with the corresponding month as the key, was stored for future aggregation.

2.3.5.3 Weekly Data & Synthesizing Features

In this part, we calculated the percentage distribution of purchases between weekdays and weekends for a specific monthly period. First of all, we added a column to the dataset indicating the day of the week for each transaction. After incorporating values for days with no transactions, we differentiated between weekdays and weekends. Finally, we converted these counts into percentage format relative to the total number of purchases and returned the result.

Then, we process monthly datasets created in step 2.3.4.2 and synthesize the required features. Initially, we synthesize features related to product variety and spending per purchase. Subsequently, we calculate the weekday-weekend distribution and

aggregate the values grouped by purchases. We then group these values by customer ID and compute the following features:

- Frequency : Total number of purchases
- Monetary : Total amount spent
- ItemVariety : Total number of distinct products purchased
- TotalItems : Total number of items purchased
- LastPurchaseDate : Date of the most recent purchase
- FirstPurchaseDate : Date of the first purchase
- StoreVariety : Number of distinct stores visited
- AverageDiscountRatio : Average discount ratio utilized
- TotalDiscountAmount : Total discount amount utilized
- PrevTotalMonetary : Total amount spent up to the current month
- PrevTotalFrequency : Total number of purchases up to the current month
- PrevItemVariety : Number of distinct products purchased up to the current month

After synthesizing these features, we merge them with previously calculated features, integrating them for further analysis. Then, the result's columns are renamed according to the date associated with the data, for example, January 2022 data was renamed as "Frequency-2022-01-01". Subsequently, the data is saved on a yearly basis to facilitate memory management. As a result, we obtain CSV files named `processed_sales_2022`, `processed_sales_2023`, and `processed_sales_2024`, which was used in the second phase of the preprocessing process.

2.3.5.4 Splitting data for Memory management

At this stage, we first need to load the results obtained from the previous step into memory and remove any erroneous values. Next, to perform operations for each user, we need to obtain a dataset containing all users. Lastly, all we need to do is group and divide the data, then save it. To achieve this, we separate the data into ten files via reading them with the necessary chunk size value.

2.3.6 Second phase of preprocessing

2.3.6.1 Extracting needed columns

At this stage, since we had recorded the values for each monthly period of our dataset in previous steps, we need a function that provides us the names of the columns we need when we input the user's reference date. This function returns the names of the columns containing the required data for the specified month and the three preceding months, which helped us to extract necessary information for each user in further processing.

2.3.6.2 Processing and Formatting Customers Data

At this stage, we read the data that is distributed into pieces on previous stages in chunks and convert date variables from string format to date-time format. We then grouped the dataset by customer ID and processed each customer individually.

On this part, via obtaining the necessary column names, we filtered the data in the necessary column names belonging to the users, and formatted these columns as "MonthlyMetric-1, -2, -3, -4" by renaming them. For example: "Frequency-1" represents the number of purchases made in the one-month period before the reference date. The algorithm retrieves the 4-month data, renames it, and combines them. Additionally, the algorithm also calculates the subscription duration of the customer until the current reference date and stores it as “SubscriptionDuration-X”, where “X” represents the month period.

Then, the purchases in the 6-month period after the current reference date are filtered, and a "churn" value is assigned to the user according to whether the user made a purchase in this period.

Afterwards, the synthesized data according to the total of purchases made before the reference date is added to the data, and finally, in order to simplify the data and save memory, we take the averages of the “StoreVariety and Weekday-Weekend” values calculated separately for 4 months and record them. The program does these calculations for each user until the end.

Lastly, upon completion of the loops, the algorithm combines all results and filters out data with no purchases over a 4-month period. The final result is then saved under the name `final_dataset`. With these developments, we are nearing the end of the preprocessing phase. In the subsequent step, we used the processed dataset to develop and test a model with the machine learning algorithm that yields the most suitable results for our dataset.

2.4 Model Development

In this section, we focused on the development of predictive models to identified customer churn for the company. The goal is to build and validate various machine learning models that can accurately predict which customers are likely to churn. Given the imbalanced nature of the dataset, it is important to apply some techniques to handle class imbalance and ensure that the models can generalize well to new data.

We began by eliminating the imbalance in the predicted classes and normalizing the dataset, followed by preparing the data for model training. Subsequently, multiple machine learning algorithms—including Random Forest, Logistic Regression, K-Nearest Neighbor, AdaBoost, Extra Tree, and Decision Tree, MLP Classifier—were employed to determine the most suitable model for our churn prediction task.

The following subsections outline the steps taken to select the best model, including the evaluation metrics and the processes used to fine-tune each algorithm.

2.4.1 Eliminate imbalance in predicted classes

The final dataset contains 81,677 unique customer values, 1,110,964 entries correspond to the churn 0 class, while 57,158 correspond to the churn 1 class. This imbalance could significantly mislead the accuracy metrics of our model and prevent us from obtaining satisfactory results. Therefore, we retain the first entries for unique customers and add all entries belonging to the churn 1 class, providing us with a more balanced training dataset. When this method was implemented, we received 62782 churned and 57158 non-churned customers.

2.4.2 Normalizing the dataset

The first filtering was performed based on the assumption that a user would not make more than 30 transactions within a single month. Even if a user were to shop every day of that month, the average frequency value should not exceed 30. By this filtering, 1915 entries are filtered.

Second filtering applied to the *Monetary* attribute was derived by visualizing the distribution of the amount spent within a month across different users, and filtering out values where the number of transactions falls below the average range of 10 to 100. After this filtering, 247 entries are filtered.

2.4.3 Preparing the dataset for training

To prepare our dataset for modeling, the first step involves creating a dataset where the target column, which the model needs to predict, has been removed. Additionally, we needed to generate a separate dataset that contained only the values of the target column, which was later used to evaluate the model's accuracy.

After creating these datasets, we used the `train_test_split` function to appropriately format the data. To further enhance accuracy, we then applied BorderLineSMOTE to oversample the data.

2.4.4 Choosing the Model

To evaluate our model for churn prediction, below listed machine learning algorithms were used:

- Random Forest
- Logistic Regression
- K Nearest Neighbor
- AdaBoost
- ExtraTree
- Decision Tree
- MLP Classifier

By using these algorithms with the prepared dataset, we were able to compare the accuracy results and select the best model fit for our purpose. Also, by the confusion matrices, it was easier to compare these algorithm's prediction power on different classes. To optimize each of the machine learning algorithms listed, GridSearch was utilized to identify the best parameters for each model. GridSearch systematically tests different combinations of hyper parameters within a specified range, allowing us to find the optimal configuration for our dataset. By applying this method, we aimed to maximize the performance of each model. The results obtained from GridSearch were used to evaluate and compare the performance of each algorithm, helping us to select the most suitable model for our churn prediction task.

The range of parameters we have searched alongside the algorithms are listed below:

- Random Forest = 'max_depth': [10,15, 20], 'min_samples_split': [2, 5, 8, 11], 'min_samples_leaf': [2,5,8,11]
- Logistic Regression = 'C': [1-12], 'penalty': ['l2', 'elasticnet', 'none'], 'class_weight': ['balanced','None']
- KNN = 'n_neighbors': [1-20]
- Ada Boost = 'n_estimators': n_estimators_values.tolist(), 'learning_rate': [0.01, 0.1, 1]
- ExtraTree = n_estimators': (50,60,70,…150)
- Desicion Tree = min_samples_leaf': [1-6], 'min_samples_split': [1-6]
- MLP Classifier = 'hidden_layer_sizes': [(50,), (100,), (100, 50), (150, 100, 50)], 'activation': ['tanh', 'relu'], 'solver': ['sgd', 'adam'], 'alpha': [0.0001, 0.05], 'learning_rate': ['constant', 'adaptive']

In the following section, we present the results of each individual algorithm`s performance and the hyper parameters that performed best for each model.

2.4.4.1 Random Forest

Best parameters of the model:

Max_depth = 20, min_samples_leaf = 2, min_samples_split = 2

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.85	0.89	0.87	12284
1 (Churned)	0.88	0.83	0.85	11272

Table 2.1: Class spesific accuracy of Random Forest

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.86	23556
Macro Avg	0.86	0.86	0.86	23556
Weighted Avg	0.86	0.86	0.86	23556

Table 2.2: Overall accuracy of Random Forest

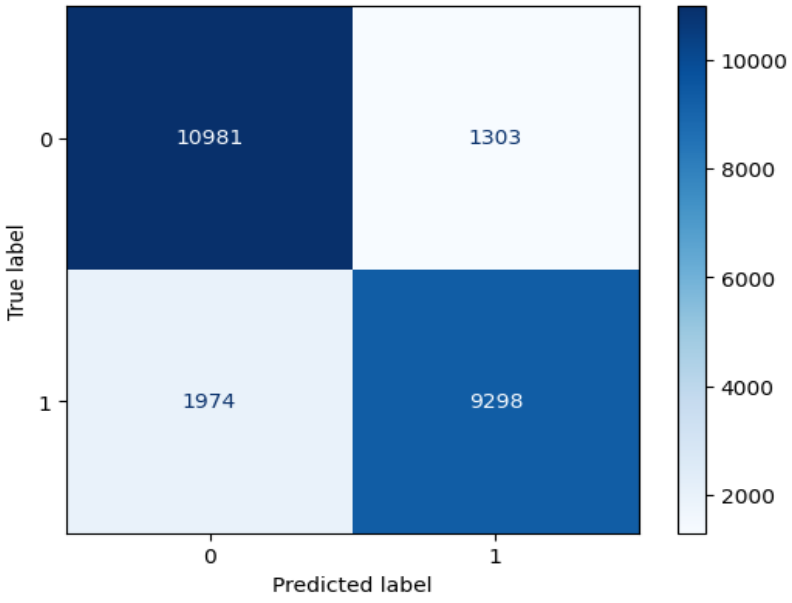

Figure 2.12: Confusion matrix of Random Forest

2.4.4.2 Logistic Regression

Best parameters of the model:

C = 7, class_weight = balanced, penalty = l2

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.71	0.86	0.78	12284
1 (Churned)	0.88	0.83	0.85	11272

Table 2.3: Class spesific accuracy of Logistic Regression

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.75	23556
Macro Avg	0.76	0.74	0.74	23556
Weighted Avg	0.75	0.75	0.74	23556

Table 2.4: Overall accuracy of Logistic Regression

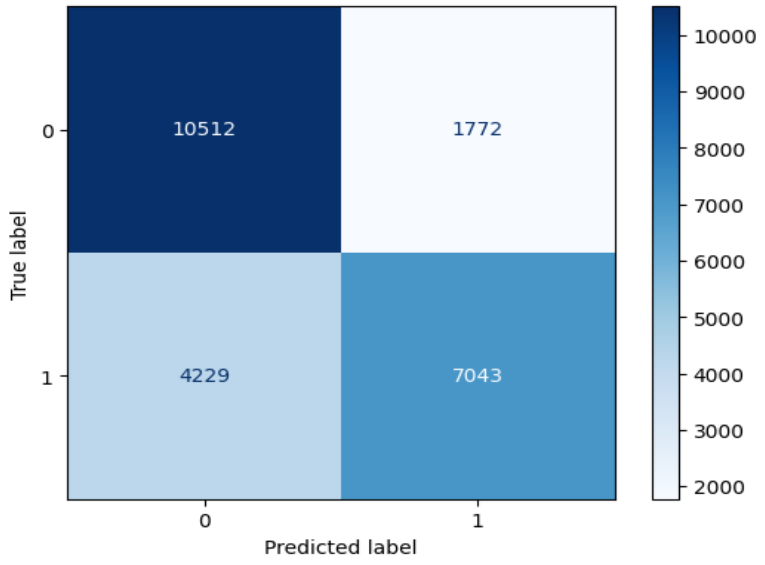

Figure 2.13: Confusion matrix of Logistic Regression

2.4.4.3 K Nearest Neighbor

Best parameters of the model:

n_neighbors = 1

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.79	0.77	0.78	12284
1 (Churned)	0.76	0.77	0.77	11272

Table 2.5: Class spesific accuracy of KNN

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.86	23556
Macro Avg	0.77	0.77	0.77	23556
Weighted Avg	0.77	0.77	0.77	23556

Table 2.6: Overall accuracy of KNN

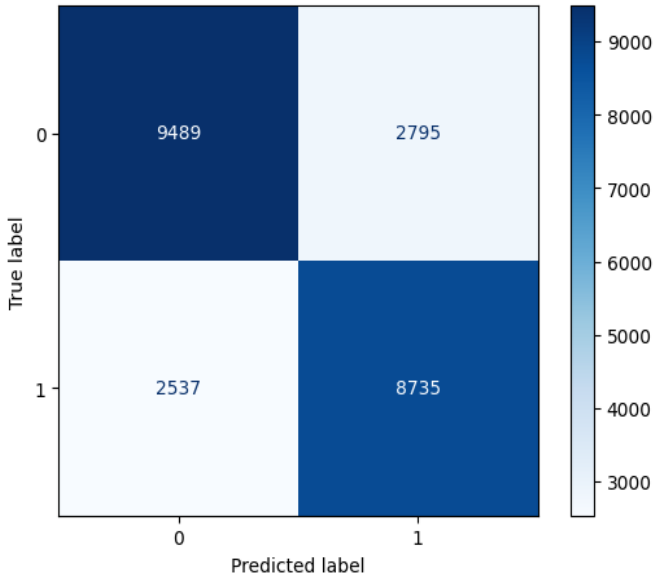

Figure 2.14: Confusion matrix of KNN

2.4.4.4 AdaBoost

Best parameters of the model:

learning_rate = 1, n_estimators = 117

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.85	0.88	0.86	12284
1 (Churned)	0.86	0.83	0.85	11272

Table 2.7: Class spesific accuracy of AdaBoost

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.86	23556
Macro Avg	0.86	0.86	0.86	23556
Weighted Avg	0.86	0.86	0.86	23556

Table 2.8: Overall accuracy of AdaBoost

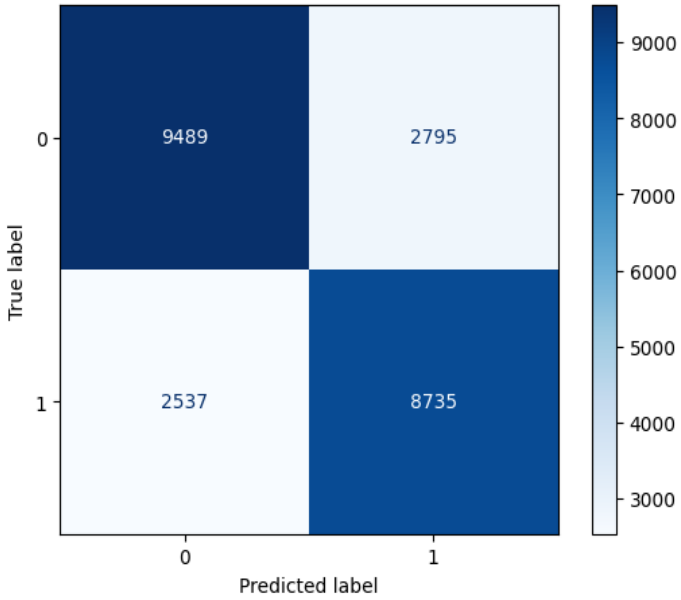

Figure 2.15: Confusion matrix of AdaBoost

2.4.4.5 Extra Tree Classifier

Best parameters of the model:

n_estimators = 130

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.85	0.89	0.87	12284
1 (Churned)	0.88	0.83	0.85	11272

Table 2.9: Class spesific accuracy of Extra Tree Classifier

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.86	23556
Macro Avg	0.86	0.86	0.86	23556
Weighted Avg	0.86	0.86	0.86	23556

Table 2.10: Overall accuracy of Extra Tree Classifier

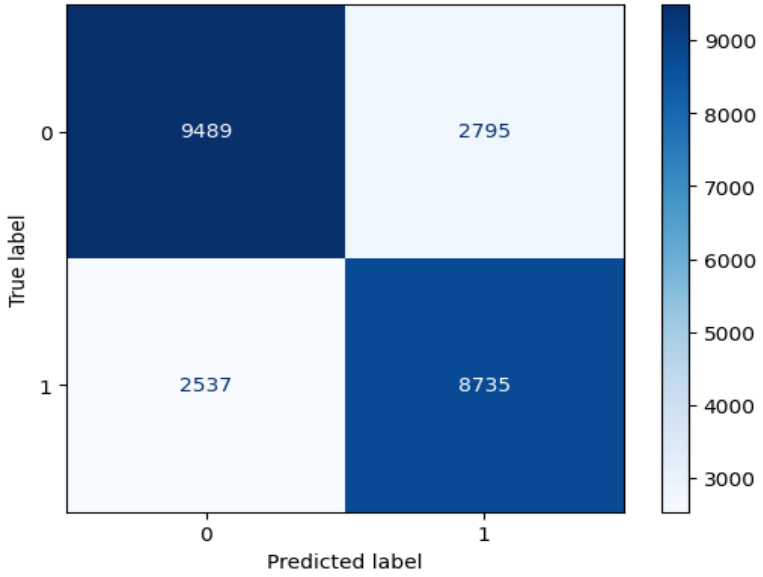

Figure 2.16: Confusion matrix of Extra Tree Classifier

2.4.4.6 Decision Tree

Best parameters of the model:

min_samples_leaf = 5, min_samples_split = 2

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.82	0.83	0.82	12284
1 (Churned)	0.81	0.80	0.80	11272

Table 2.11: Class spesific accuracy of Decision Tree

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.81	23556
Macro Avg	0.81	0.81	0.81	23556
Weighted Avg	0.81	0.81	0.81	23556

Table 2.12: Overall accuracy of Decision Tree

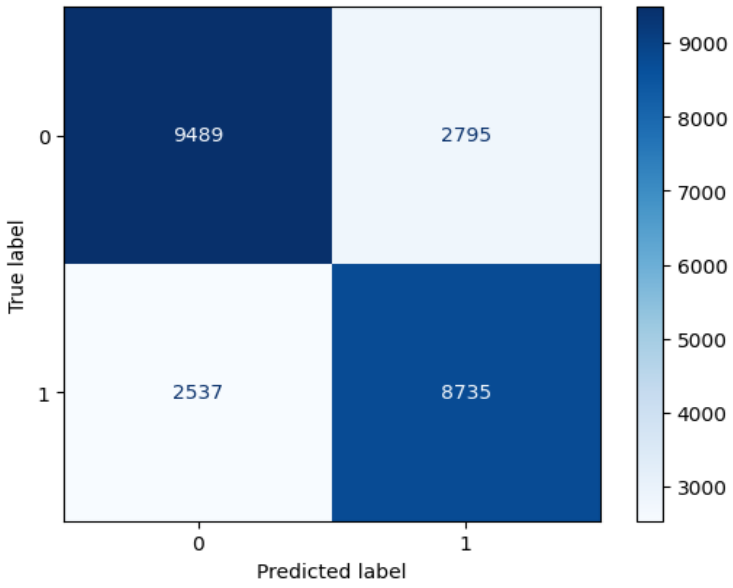

Figure 2.17: Confusion matrix of Decision Tree

2.4.4.6 MLP Classifier

Best parameters of the model:

Activation = relu, alpha = 0.05, hidden_layer_sizes = (100,50), learning_rate = constant, solver = adam

Category	Precision	Recall	F1-Score	Support
0 (Not Churned)	0.85	0.90	0.87	12284
1 (Churned)	0.88	0.83	0.85	11272

Table 2.13: Class spesific accuracy of MLP Classifier

Category	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.86	23556
Macro Avg	0.86	0.86	0.86	23556
Weighted Avg	0.86	0.86	0.86	23556

Table 2.14: Overall accuracy of MLP Classifier

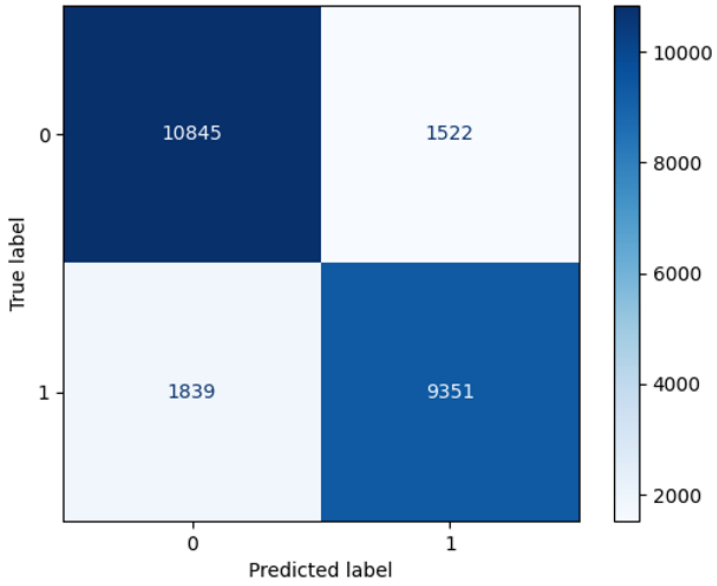

Figure 2.18: Confusion matrix of MLP Classifier

2.4.4.7 Final Evaluation and Conclusion

Upon evaluating all the chosen models, it was found that the Random Forest and MLP Classifier algorithm achieved the highest accuracy, outperforming the other models in predicting customer churn. The Random Forest and MLP Classifier model not only excelled in overall accuracy but also demonstrated a balanced performance across various metrics, as reflected in the classification report and confusion matrix. This suggests that the model is effective in distinguishing between customers who are likely to churn and those who are not. Given its robustness and consistency across different evaluation metrics, the Random Forest model stands out as the most suitable choice for our churn prediction task. This model was used in the final deployment phase to identify at-risk customers, enabling targeted retention strategies.

3. DISCUSSION

This study highlights several important aspects of applying machine learning techniques for customer churn prediction for the company. Beyond the immediate results, several key points emerge that are crucial for interpreting the findings and guiding future strategies.

Firstly, the effectiveness of machine learning models like Random Forests, MLP Classifier, Extra Trees, and AdaBoost demonstrated their potential for enhancing customer retention strategies for the company. However, the successful implementation of these models requires ongoing adaptation and optimization. The retail environment is dynamic, with customer behaviors continually evolving. Therefore, the models need to be permanently updated with new data to account for shifts in customer behavior and market trends.

The study also reveals the importance of feature engineering and data enrichment in improving model performance. While the existing dataset provided a foundation for developing predictive models, its limitations—such as a lack of comprehensive customer information—suggest areas for enhancement. Incorporating additional features, such as detailed purchase histories, product preferences, and engagement metrics, could offer

deeper insights and improve model accuracy. Moreover, leveraging data from other sources, such as customer feedback or website activity, could provide valuable context to better understand the drivers of churn.

Another important consideration is the role of segmentation in model development. The analysis shows that certain customer segments, particularly those with infrequent transactions, pose a challenge to predictive accuracy due to their disproportionate representation in the dataset. Future efforts could benefit from developing more granular segmentation strategies that focus on differentiating between various customer types, such as high-value versus low-value customers or frequent versus occasional shoppers. This targeted approach could help create more tailored retention strategies that address the unique needs and behaviors of different customer groups.

Additionally, the discussion points to the importance of collaboration across departments to maximize the value of the churn prediction models. For example, input from marketing and customer service teams could provide crucial context regarding customer behavior trends, promotional effectiveness, and customer feedback. Integrating this qualitative information with quantitative model outputs can enhance the overall customer retention strategy, ensuring that it aligns closely with the company's business goals.

Finally, the findings of this study suggest that while machine learning models are a powerful tool for predicting customer churn, their utility is significantly enhanced when combined with business expertise and strategic planning. Hence, the company can benefit from a continuous feedback loop where model predictions inform business decisions, and real-world outcomes are used to refine and improve the models further.

In conclusion, while the models developed in this study offer a promising approach to understanding and mitigating customer churn, ongoing refinement, data enrichment, and cross-functional collaboration will be key to fully realizing their potential and driving sustainable business growth for the company.

CONCLUSIONS AND FURTHER WORK

This section synthesizes the findings from the data analysis and model evaluation phases, providing insights into the effectiveness of the chosen approaches in predicting customer churn. By examining the performance of various machine learning models and the implications of their results, we aim to understand the strengths and limitations of our methods, as well as their practical applications in real-world scenarios. The discussion will also address potential areas for improvement and suggest future directions for enhancing the accuracy and reliability of churn prediction models.

Evaluation of models performance

It is clear that the best fit machine learning algorithm and its model for our purposes and our dataset is the Random Forest with %86 accuracy. Even though the tree-based algorithms such as AdaBoost, Extra Tree and Decision Tree perform roughly the same performance and the Neural Network algorithm MLP Classifier performed almost the same, this slight difference is important for predicting possible churn and avoiding losing customers. The confusion matrices show this performance difference more effectively and the one with least wrong predictions on the churned customers, which is the RandomForest.

Class imbalance and segmentation

The imbalanced distribution of the churn class within the dataset has been a significant factor negatively impacting the model's performance. To address this imbalance and enhance the accuracy of the model in predicting churn, BorderlineSMOTE and synthetic sampling techniques were applied. But still, the imbalance and misleading values in some segmentations of the dataset are major problems on the model’s accuracy. Even though the %86 accuracy can be considered well-performed, it should not be forgotten that the data led to this accuracy is cleared from customers that have transactions less than 20 and monetary less than a thousand Turkish liras in 29 months. The model performs well in this segmentation of customers, but in the customers who have transactions less frequently, our model performs with %73 accuracy due to misleading

and imbalanced data, which can be considered as a reference level performance for a model operating on an imbalanced data set.

Observations and Future Work

The findings and observations provide valuable insights into company's customer churn prediction. For future work and to increase the model's performance, there are several points that need to be addressed by both the company and us. On the development side, some features in the training dataset are slightly less effective than others. Refining these features or creating new, relevant features could improve the model's performance. Additionally, using more advanced algorithms such as Gradient Boosting Machines (GBM), Support Vector Machines (SVM), is likely to enhance the model's accuracy.

Regarding the raw dataset, enrichment is clearly needed. The data could be supplemented with customer information such as age, gender, ethnicity, customer satisfaction surveys, or social media data. Enrichment is also necessary for certain customer segments; the data for customers with relatively infrequent transactions constitutes the majority of the dataset and can be misleading, thus requiring further refinement. Another issue with the dataset is that it includes not only unique customers but also physical store employees, such as cashiers. Entries related to these employees are likely to mislead our model, and filtering them out is challenging since their customer IDs are not explicitly identified. Eliminating these entries from the dataset makes the processing phase easier and likely to increase the performance of the model due to less misleading entries.

In summary, this research demonstrates the significant potential of machine learning algorithms in accurately predicting customer churn of company`s customers. Among the various algorithms evaluated, Random Forest, Extra Tree, Decision Tree, AdaBoost and MLP Classifier have emerged as particularly effective, offering robust predictive capabilities and actionable insights. These models enable companies to identify customers who are at high risk of churning, allowing for the implementation of targeted retention strategies. By intervening proactively, businesses can not only enhance customer retention but also drive overall success and growth in a highly competitive market. The

integration of such advanced analytical techniques into the company's operational processes underscores the transformative power of data-driven decision-making in maintaining a loyal customer base and sustaining business performance.

REFERENCES

[1] Equihua, Juan Pablo, et al. “Modelling Customer Churn for the Retail Industry in a Deep Learning Based Sequential Framework.” ArXiv.org, 2 Apr. 2023, arxiv.org/abs/2304.00575.

[2] Z. Wu, L. Jing, B. Wu, and L. Jin, “A PCA-AdaBoost model for E-commerce customer churn prediction,” Annals of Operations Research, vol. –, no. –, Jan. 2022. Available: https://doi.org/10.1007/s10479-022-04526-5.

[3] X. Wu and S. Meng, "E-commerce customer churn prediction based on improved SMOTE and AdaBoost," in 2016 13th International Conference on Service Systems and Service Management (ICSSSM), Kunming, China, 2016, pp. 1–5, doi: 10.1109/ICSSSM.2016.7538581.

[4] P. Nagaraj, V. Muneeswaran, A. Dharanidharan, M. Aakash, K. Balananthanan, and C. Rajkumar, "E-Commerce Customer Churn Prediction Scheme Based on Customer Behaviour Using Machine Learning," in 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1–6, doi: 10.1109/ICCCI56745.2023.10128498.

[5] I. Jahan and T. F. Sanam, "An Improved Machine Learning Based Customer Churn Prediction for Insight and Recommendation in E-commerce," in 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2022, pp. 1–6, doi: 10.1109/ICCIT57492.2022.10054771.

[6] Customer Churn Prediction for Subscription Businesses Using Machine Learning: Main Approaches and Models. Available: https://www.altexsoft.com/blog/business/customer-churn-prediction-for-subscription-businesses-using-machine-learning-main-approaches-and-models/. (Accessed: Sep. 2, 2024).

[7] H. Guliyev and F. Yerdelen, “Customer churn analysis in banking sector: Evidence from explainable machine learning models,” Journal of Applied Microeconometrics, vol. 1, no. 2, pp. 85–99, 2021.

[8] Decision Tree Classification Algorithm. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. (Accessed: Aug. 5, 2024).

[9] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms. IEEE Press, United States, 2011, p. 121.

[10] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, “Customer churn prediction system: A machine learning approach,” Computing, vol. –, pp. 1–24, 2021.

[11] F. Ehsani, “Customer churn prediction from Internet banking transactions data using an ensemble meta-classifier,” unpublished, 2022.

[12] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. Available: https://doi.org/10.1023/a:1010933404324.

[13] Y. He, Y. Xiong, and Y. Tsai, “Machine Learning Based Approaches to Predict Customer Churn for an Insurance Company,” in 2020 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2020.

[14] S. Kim, D. Choi, E. Lee, and W. Rhee, “Churn prediction of mobile and online casual games using play log data,” PLOS ONE, vol. 12, no. 7, p. e0180735, 2017.

[15] C. Mahapatra, “Analysis of MLP model on WSN localization data by using various training functions,” in 2014 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, Mar. 2016, pp. 250–252.