**Customer Churn Analysis Report**

Contents

[Introduction 3](#_Toc129557050)

[Methodology to clean the dataset 3](#_Toc129557051)

[Dropping the attributes 3](#_Toc129557052)

[Feature Engineering 4](#_Toc129557053)

[Label Encoding 4](#_Toc129557054)

[Transforming the categorical labels to numeric 4](#_Toc129557055)

[Scaling the dataset 4](#_Toc129557056)

[Model Selection 5](#_Toc129557057)

[Evaluating the models 5](#_Toc129557058)

[Results 5](#_Toc129557059)

[Implementing SVM Model: 5](#_Toc129557060)

[Results of scaled data 5](#_Toc129557061)

[Implementing Random Forest Classifier 5](#_Toc129557062)

[Results of scaled data 5](#_Toc129557063)

[Implementing Logistic Regression 5](#_Toc129557064)

[Results of scaled data 5](#_Toc129557065)

[Conclusion 6](#_Toc129557066)

# Introduction

This report outlines the development of a machine learning model to predict payment risk for online purchase orders. The goal is to help an online merchant identify customers who are likely to pay for their products. The dataset contained 30,000 online purchase orders with 44 attributes, one of which indicated a high or low risk of default payment.

# Methodology to clean the dataset

The first step in this project was to carefully analyze the attributes of the training dataset. I identified which attributes were most relevant to the classification task and preprocessed the data accordingly.

## Dropping the attributes

The following stepwise approach was taken to clean the dataset.

1. Dropping ‘ANUMMER\_01', 'ANUMMER\_02', 'ANUMMER\_03', 'ANUMMER\_04', 'ANUMMER\_05', 'ANUMMER\_06', 'ANUMMER\_07', 'ANUMMER\_08', 'ANUMMER\_09', and 'ANUMMER\_10':

**Reason:** These columns contained almost 80% or above null values. To reduce the dimensionality of the dataset these columns were dropped.

1. Imputing the null values with unknown:

Reason: The missing values in the attributes were imputed with 'unknown' to avoid misclassification of customers with incorrect card names.

1. Applying feature transformation:

Reason:

1. It was observed that the 'DATE\_LORDER' column in the dataset provides information about the customer's last order. The minimum and maximum dates of the last order in the dataset are 12th Dec 2000 and 19th Jan 2005, respectively. A new feature was proposed to be created using this information to improve the model's performance. However, almost 50% of the data, i.e., 14808 rows, have missing values in this column, and imputing these values would introduce bias in the model. Therefore, it was decided to drop the 'DATE\_LORDER' column.
2. As a part of feature engineering, the 'Z\_Card\_valid' attribute was split into two new attributes, 'Valid\_date' and 'Valid\_month'. This was done to extract more meaningful features from the original attribute, which contains the expiration date of the customer's credit card.
3. Imputing the columns containing unknown values and replacing with ‘9’.

Reason: 'MAHN\_AKT' attribute and 'MAHN\_HOECHST' attribute appears to contain a code for the current stage of reminders, with values ranging from 0 to 3. Missing values in this attribute likely represent reminder stages that are not available. To handle these missing values, it was imputed with a value of 9, representing 'not available/unknown'. This imputation approach was likely taken to avoid losing valuable information while also preventing bias in the model.

1. Dropping the columns:
2. Birthdate: 'Birthdate' column was dropped instead of imputing missing values due to the high number of unique values, which could introduce bias in the model.
3. Time\_orders: The 'TIME ORDER' column has only 20 missing values, and this column can provide critical information about rejected orders. As a result, it was decided to drop the null records and work with the data that was available. Because the number of null records is small, it is unlikely to have a significant impact on the model.
4. ORDER\_ID: this column contains unique values that identify each order, but it does not provide any useful information for classification. Therefore, it was decided to drop this column from the dataset.

## Feature Engineering

Applying feature engineering on columns:

1. Converting the Time\_Order Column into Hours and Minutes
2. Converting all the text columns to numbers

## Label Encoding

There are several columns in the data set that have Yes/No values. The "label encoding" technique was used to handle these columns. After these columns have been encoded, the other columns will be evaluated to determine the best method for handling them.

The following code was used:

def impute\_label(df, col\_name):

  df[col\_name] = df[col\_name].map({'yes':1 ,'no':0})

The following columns were imputed:

'CLASS', 'B\_EMAIL', 'B\_TELEFON', 'FLAG\_LRIDENTISCH', 'FLAG\_NEWSLETTER', 'Z\_METHODE', 'Z\_CARD\_ART', 'Z\_LAST\_NAME', 'CHK\_LADR', 'CHK\_RADR', 'CHK\_KTO', 'CHK\_CARD', 'CHK\_COOKIE', 'CHK\_IP', 'FAIL\_LPLZ', 'FAIL\_LORT', 'FAIL\_LPLZORTMATCH', 'FAIL\_RPLZ', 'FAIL\_RORT', 'FAIL\_RPLZORTMATCH', ‘NEUKUNDE','MAHN\_AKT', 'MAHN\_HOECHST', 'V\_MONTH', 'V\_YEAR'

## Transforming the categorical labels to numeric

The 'Z\_METHODE' column in the 'df\_cleaned\_train' dataframe is being mapped to a dictionary that associates the string values 'check', 'credit\_card', 'debit\_note', and 'debit\_card' with the numerical values 1, 2, 3, and 4, respectively. This process was repeated for 'Z\_CARD\_ART', WEEKDAY\_ORDER, 'Z\_LAST\_NAME'

# Scaling the dataset

To prepare the data for training the classification model, I first normalized the numerical features in the dataset using standard scaling. This was done to ensure that the features were on a similar scale, which can aid the model's convergence and overall performance.

After scaling the data, I divided it into two sets: one with the features (X) and one with the target variable (y). These sets were then used to train the classification model using various machine learning algorithms.

Along with training the model on scaled data, I ran the same algorithms on unscaled data to compare their performance. This enabled me to determine whether scaling the data had any effect on the model's accuracy and other metrics.

# Model Selection

I tested three popular machine learning models for this project: Support Vector Machine (SVM), Random Forest, and Logistic Regression. On the preprocessed and scaled dataset, I trained each of these models and evaluated their performance using several metrics such as accuracy, precision, recall, and F1 score. I was able to determine which model performed the best for the given dataset problem by comparing the results of these models.

## Evaluating the models

I evaluated the models' performance using a variety of metrics, including accuracy, precision, recall, and F1 score. I was able to determine which algorithms performed best and whether scaling the data improved their performance by comparing the results of models trained on scaled and unscaled data.

# Results

## Implementing SVM Model:

### Results of scaled data

The SVM model achieved an accuracy of 0.94 with a precision of 0.94 and a recall of 1. The F1 score was calculated to be 0.97, indicating a good balance between precision and recall. However, the model had a relatively low AUC value of 0.59, suggesting that it may not perform as well in scenarios with imbalanced classes.

To further analyze the performance of the model, the cost matrix was used to see the cost of misclassifying high-risk and low-risk customers. The cost matrix had values of 5639 for correctly classified instances and 0 for misclassifying a high-risk customer as low-risk and 357 for misclassifying a low-risk customer as high-risk.

Overall, the SVM model performed reasonably well in predicting the risk level of customers. However, it was not accurate to predict high risk customers based on cost matrix and AUC value.

## Implementing Random Forest Classifier

### Results of scaled data

Training a Random Forest classifier on the data. This model achieved an accuracy score of 0.936 and an AUC score of 0.664. The cost of predicting a low-risk customer as high-risk customer is 37, while the cost of predicting a high-risk customer as low-risk customer is 347.

This model performed slightly worse than the SVM model, as it predicted 37 customer as a high-risk customer but still performed well in predicting high-risk customers with a low number of false negatives.

## Implementing Logistic Regression

### Results of scaled data

The model achieved an accuracy of 0.940, which means that it correctly classified 94% of the instances in the test set. The precision for the positive class is only 0.50, indicating that when the model predicts a positive instance, it is correct only half of the time. The recall for the positive class is only 0.01, indicating that the model is missing a large number of low-risk customers. The cost matrix indicates that the model correctly classified 5636 high-risk as high-risk customers, incorrectly classified 3 high-risk as low-risk, incorrectly classified 354 low-risk as high-risk, and correctly classified only 3 low-risk customers.

The AUC value of 0.735 suggests that the model is performing slightly better than random classifier and SVM. Overall, the report indicates that while the model has a high accuracy, it is not performing well for the positive class. But due to chances of predicting high risk customers accurately, this model is selected to run the test dataset.

# Conclusion

The scaled-data performed better in the model than the unscaled data. Thus, scaling was used in the test data. All the data preprocessing was applied to test dataset.

Following observations were taken from the models:

1. SVM model has an accuracy of 0.94 and correctly classified all high-risk instances but failed to classify any low-risk customers. The precision and recall for the low-risk are both zero, indicating that the model is not able to identify any low-risk default payment. The AUC value for the SVM model is 0.587, which is the lowest among the three models.
2. Random Forest Classifier has an accuracy of 0.94 and performed better than the SVM model for the low-risk with a precision of 0.21 and a recall of 0.03. However, it still has poor performance for the low-risk with an F1-score of 0.05. The cost matrix shows that the model correctly classified 5602 high-risk instances and 10 low-risk instances, but it incorrectly classified 347 lo-risk instances as high-risk customers and 37 high-risk as low-risk customers. The AUC value for the Random Forest Classifier model is 0.668, which is better than the SVM model.
3. Logistic Regression model has an accuracy of 0.94 and similar to the Random Forest Classifier, it performed better than the SVM model for the low-risk class with a precision of 0.50 and a recall of 0.01. However, the performance of the Logistic Regression model for the low-risk class is still not good enough with an F1-score of 0.02. The cost matrix shows that the model correctly classified 5636 high-risk customers and 3 low-risk customers, but it incorrectly classified 354 low-risk as high-risk and 3 high-risk instances as low-risk. The AUC value for the Logistic Regression model is 0.735, which is the highest among the three models.

In conclusion, all three models have high accuracy, but they are not performing well for the positive class. The Logistic Regression model performed better than the other two models based on the AUC value. Hence, Logistic Regression was used as the model on test dataset to predict the class labels in the file output.