**Calculate and predict Customer Lifetime Value**

1. Calculate and predict Customer Lifetime Value (CLV).
   1. Calculate CLV using different approaches and frameworks.
   2. Explore predictive modeling techniques such as linear regression and logistic regression for CLV prediction.
   3. Assess the accuracy and reliability of CLV predictions.

**Theory:**

Predicting Customer Lifetime Value (CLV) involves estimating the total revenue a business can expect to earn from a customer throughout their entire relationship. CLV is crucial for businesses to make informed decisions about marketing strategies, customer acquisition costs, and overall business profitability.

**CLV formula:**

Customer Value = Average Purchase Value x Average Number of Purchases Customer Lifetime Value = Customer Value x Average Customer Lifespan **Implementation Process**

1. Discuss whether CLV fits as a metric in our business
2. Identification and understanding of sources and metadata
3. Extract, transform, clean and load data
4. Choose CLV method
5. Analyze results and adjust parameters
6. Present and explain the results

**Code:**

**# Step1:Install and Load Required Packages:**

# Load required libraries install.packages(c("dplyr", "tidyr", "caret")) library(dplyr)

library(tidyr) library(caret)

**#Step2:Import and Explore Data:**

# Read the file into a dataframe

df <- read.csv("D:/MSc DS/Semester 1/Retail Market Analysis/Practical/bank.csv")

**#Step3: Explore the dataset**

summary(df) str(df)

**#Step3: Data Preprocessing** any\_missing <- any(is.na(df)) print(any\_missing)

df$default <- as.factor(df$default) df$housing <- as.factor(df$housing) df$loan <- as.factor(df$loan)

df <- df[, !(names(df) %in% c("unnecessary\_variable1", "unnecessary\_variable2"))] boxplot(df$balance, main = "Balance Boxplot", ylab = "Balance")

z\_scores <- scale(df$balance) threshold <- 3

outliers <- abs(z\_scores) > threshold

df <- df[!outliers, ]

**#Step4: Split the dataset into training and testing sets**

set.seed(123)

#Sets the seed for reproducibility in generating random numbers or sequences. split\_index <- createDataPartition(df$duration, p = 0.8, list = FALSE)

# Splits the 'duration' column of dataframe 'df' into training (80%) and testing (20%) indices without creating a list. train\_data <- df[split\_index, ]

#Assigns the rows indexed by 'split\_index' to the 'train\_data' dataframe for training. test\_data <- df[-split\_index, ]

# Assigns the rows not in 'split\_index' to 'test\_data' for model evaluation. #Why we do this?

#Splitting the dataset ensures an unbiased assessment of a model's performance by training on one portion and testing #on another, guarding against overfitting and gauging its generalizability to new data, aiding in hyperparameter tuning #for better predictions on unseen information.

**#Step5: Linear regression model for CLV prediction**

model <- lm(duration ~ age + balance + campaign + pdays + previous, data = train\_data)

#Creates a linear regression model ('model') predicting 'duration' based on 'age', 'balance', 'campaign', 'pdays', and #'previous' using training data.

# Make predictions on the test set

predictions <- predict(model, newdata = test\_data)

#Assess the accuracy and reliability using Mean Squared Error(MSE) for linear regression mse <- mean((test\_data$duration - predictions)^2)

print(paste("Mean Squared Error (MSE):", mse))

#Note: Calculates the Mean Squared Error (MSE) by computing the average squared differences between the actual #'duration' values in the test data and the predicted values.

#Prints the MSE value, quantifying the average squared deviation between predicted and actual durations, assessing #the model's prediction accuracy.

#Visualize the predicted Vs Actual values (linear regression):

ggplot() +

geom\_point(aes(x = test\_data$duration, y = predictions), color = "blue") + geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") + labs(title = "Actual vs. Predicted CLV", x = "Actual CLV", y = "Predicted CLV")

# It creates a scatter plot comparing actual CLV values from the test data with predicted CLV values using blue points. #The red dashed line represents perfect prediction where actual equals predicted CLV, aiming to visualize how close #the predictions are to the actual values in a simple, clear manner.

**#Step6: Logistic Regression model for CLV predictions**

logistic\_model <- glm(default ~ age + balance + campaign + pdays + previous, data = train\_data, family = "binomial")

# Creates a logistic regression model ('logistic\_model') to predict the 'default' variable based on 'age', 'balance', #'campaign', 'pdays', and 'previous' predictors using training data, assuming a binomial distribution for the response #variable.

# Make predictions on the test set

logistic\_probabilities <- predict(logistic\_model, newdata = test\_data, type = "response") # Convert probabilities to binary predictions (0 or 1)

logistic\_predictions <- ifelse(logistic\_probabilities > 0.5, 1, 0)

#Converts predicted probabilities ('logistic\_probabilities') into binary predictions (0 or 1) by setting a threshold of 0.5: #assigning 1 if the probability is above 0.5 (indicating a positive outcome) and 0 otherwise (representing a negative #outcome), simplifying classification based on probabilities.

#Assess the accuracy and reliability using Mean Squared Error(MSE) for logistic regression: logistic\_accuracy <- sum(logistic\_predictions == test\_data$default) / nrow(test\_data) print(paste("Logistic Regression Accuracy:", logistic\_accuracy))

# Calculates the accuracy of the logistic regression model ('logistic\_accuracy') by comparing predicted binary outcomes #('logistic\_predictions') with the actual 'default' values in the test data and dividing by the total number of observations, #then prints the accuracy as a proportion.

**#Step7: Visualize the predicted Vs Actual values (logistic regression)**

ggplot() +

geom\_point(aes(x = test\_data$default, y = logistic\_probabilities), color = "blue") + geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +

labs(title = "Actual vs. Predicted CLV (Logistic Regression)", x = "Actual CLV", y = "Predicted CLV")

**Interpretation and Implications:**

The linear regression model yielded a Mean Squared Error (MSE) of 120826.42, indicating the average squared difference between actual and predicted CLV values. The logistic regression model had an accuracy of 0, suggesting that the predictions did not match the actual outcomes. Possible reasons include class imbalance, feature selection, or data quality issues.

Further analysis and model refinement may be necessary to improve the predictive performance of both models. This practical aimed to demonstrate the application of linear and logistic regression for Customer Lifetime Value (CLV) prediction, emphasizing the importance of careful data preprocessing and model evaluation.