

R Codes:

```
# R code for the project
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(caret)
library(randomForest)

# Load the dataset
data <- read.csv("customer_data.csv") # Replace with your dataset file path

# Inspect the dataset
head(data)
summary(data)

# Data Preprocessing
# Convert categorical variables if any
data$Customer_ID <- as.factor(data$Customer_ID)

# Check for missing values
sum(is.na(data))

# Handle missing values if necessary (imputation or removal)
data <- na.omit(data)

# Feature Engineering (if applicable)
# Add interaction terms, scale variables, etc.
data$Engagement_Score <- data$Email_Opens * data$Purchase_Frequency

# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(data$CLV, p = 0.8, list = FALSE)
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]

# Train a Random Forest Model
set.seed(123)
rf_model <- randomForest(CLV ~ Purchase_Frequency + Total_Spend + Email_Opens + Age
+ Engagement_Score,
                        data = trainData, importance = TRUE, ntree = 500)

# Evaluate model performance on test data
predictions <- predict(rf_model, testData)
postResample(predictions, testData$CLV)

# Variable Importance
varImpPlot(rf_model)

# Visualize Relationships
# CLV vs. Purchase Frequency
ggplot(data, aes(x = Purchase_Frequency, y = CLV)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  theme_minimal() +
  labs(title = "CLV vs. Purchase Frequency", x = "Purchase Frequency", y = "CLV")

# CLV by Loyalty Program Participation
data$Loyalty_Program <- as.factor(ifelse(runif(nrow(data)) > 0.5, "Yes", "No")) #
```

```

Simulated for example
ggplot(data, aes(x = Loyalty_Program, y = CLV, fill = Loyalty_Program)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "CLV by Loyalty Program Participation", x = "Loyalty Program", y =
"CLV")

# CLV vs. Engagement Score
ggplot(data, aes(x = Engagement_Score, y = CLV)) +
  geom_point() +
  geom_smooth(method = "lm", col = "green") +
  theme_minimal() +
  labs(title = "CLV vs. Engagement Score", x = "Engagement Score", y = "CLV")

# Save the model for future use
saveRDS(rf_model, "rf_model.rds")

# Load the model for inference
loaded_model <- readRDS("rf_model.rds")

# Predictions for new data
new_data <- data.frame(Purchase_Frequency = c(5, 3), Total_Spend = c(500, 300),
                        Email_Opens = c(20, 15), Age = c(35, 30), Engagement_Score =
c(100, 45))
predict(loaded_model, new_data)

```

Explanation:

1. Data Loading and Inspection: Reads the dataset and provides an overview.
2. Preprocessing: Handles missing values, creates new features, and converts data types.
3. Model Training: Implements a Random Forest model for CLV prediction.
4. Evaluation: Tests model performance using the caret package and plots variable importance.
5. Visualization: Creates plots to visualize relationships between features and CLV.
6. Saving and Loading Models: Demonstrates how to save and reuse the trained model.
7. Predictions: Uses the model to predict CLV for new customer data.