Anticipating Customer Attrition: A Predictive Model Approach

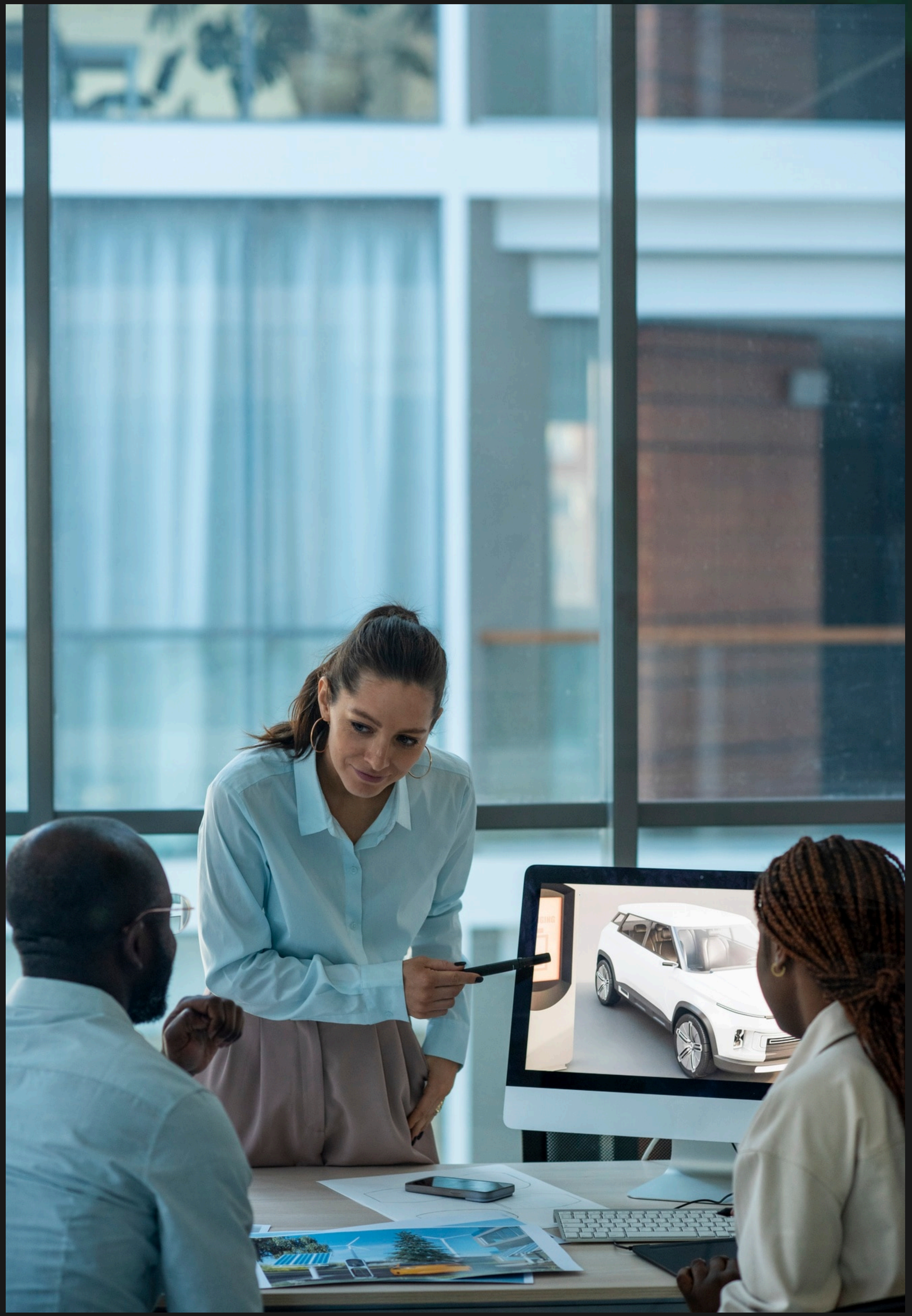
Presented by: The Mode-rn Thinkers

Overview

Intro: This project focuses on predictive modeling to forecast customer churn in a subscription-based service.

Tools and Libraries: Utilizing R programming language, we leverage various libraries including caret, gbm, pROC, and others for data preprocessing, model building, and evaluation.

Model: The analysis led to the selection of the Gradient Boosting Model, which was the most optimized for the Churn data provided presenting us with an **AUC of 0.753**



Data Exploration & Feature Engineering

Data Load and Summary: We load the training dataset and perform preliminary exploration using summary statistics and structure inspection.

Feature Engineering: Employed feature engineering techniques to enhance predictive power including ratio computation, combination of features, and creation of new categorical variables.

```r
# Feature engineering

# Feature for the Ratio of "daysInactive" to "activeSince"
data$inactive_ratio <- data$daysInactive / data$activeSince

# Combine "daysInactiveAvg" and "daysInactiveSD" into a Single Feature
data$inactive_variability <- data$daysInactiveAvg + data$daysInactiveSD

# Combine "productCategories" and "productViews" into a Single Feature for Average Diversity
data$avg_product_diversity <- (data$productViews + data$productCategories) / 2

# Create a New Feature by Categorizing "timeOfDay"
# Assuming "timeOfDay" is in POSIXct format
data$time_of_day <- cut(data$timeOfDay, breaks = c(-Inf, 6, 12, 18, Inf), labels = c("Night", "Morning", "Afternoon", "Evening"), include.lowest

# Combine "duration" with "visits" to Create a Feature for Total Time Spent
data$total_time_spent <- data$duration * data$visits

# Sum Clicks Across Different Product Categories to Create an Overall Engagement Score
data <- data %>%
  group_by(id) %>%
  mutate(overall_engagement_score = sum(clicks, na.rm = TRUE))

# Group Similar Product Categories to Reduce Dimensionality
category_mapping <- list(
  Fashion = c("clicksClothing", "clicksShoes"),
  Tech = c("clicksElectronics", "clicksWatches"),
  Literature = c("clicksBooks", "clicksMovies", "clicksMusic"),
  Home = c("clicksKitchen", "clicksHome", "clicksGarden", "clicksPet", "clicksFood"),
  Toys = c("clicksToys"),
  Tools = c("clicksTools", "clicksAutomotive", "clicksOutdoors", "clicksHandmade", "clicksSports", "clicksScience", "clicksIndustrial")
)
```

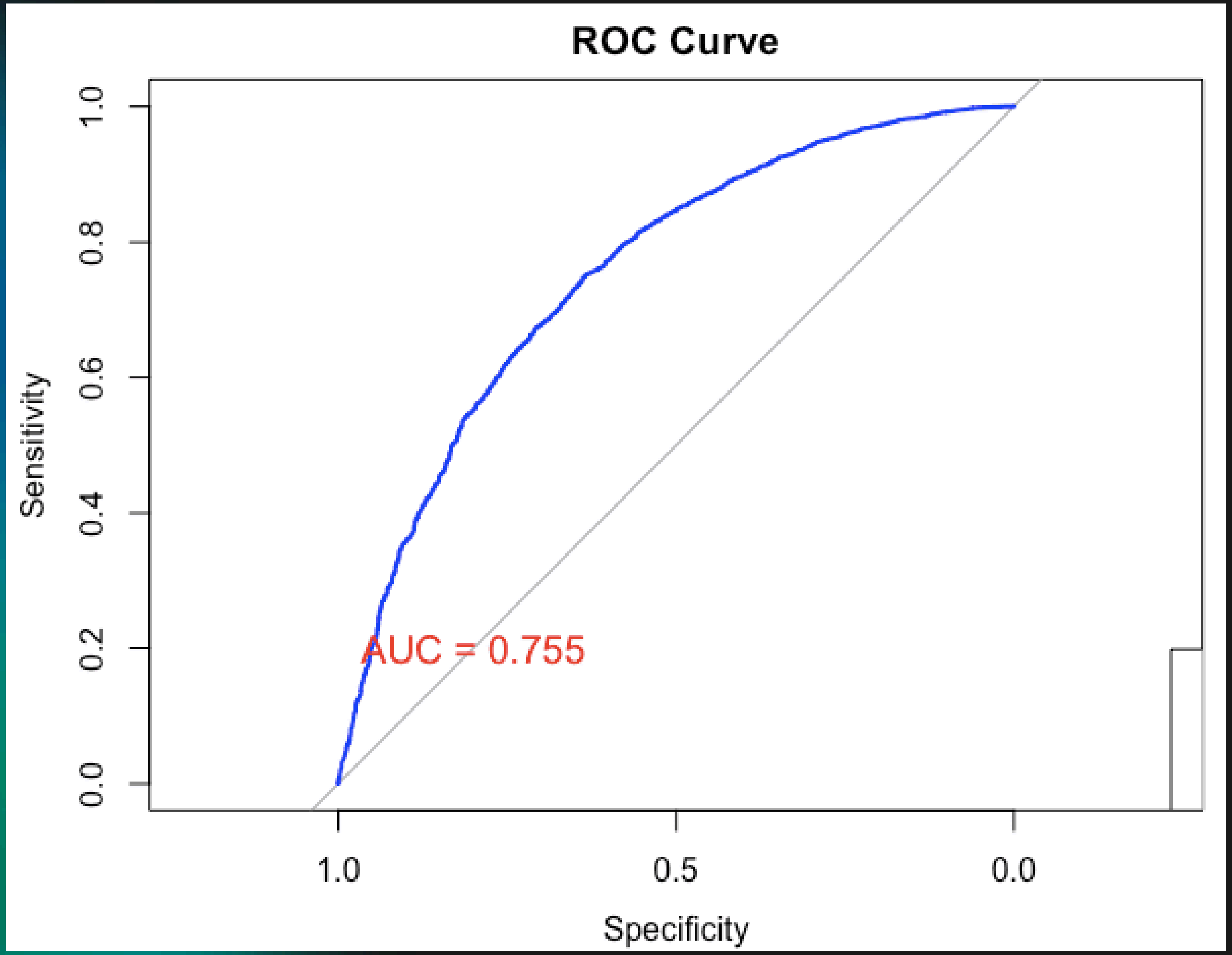
```r
################## MODEL ###################

# Split the data into training and testing sets

library(caret)
library(pROC)

set.seed(123)
train_index <- createDataPartition(data$churn, p = 0.6, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Train the GBM model
gbm_model <- gbm(churn ~ ., data = train_data, distribution = "bernoulli", n.trees = 100, interaction.depth = 4)

# Make predictions on the testing set
predictions <- predict(gbm_model, newdata = test_data, n.trees = 100, type = "response")

# Convert predicted probabilities to binary predictions
binary_predictions <- ifelse(predictions > 0.5, 1, 0)

# Evaluate performance using confusion matrix
conf_matrix <- table(test_data$churn, binary_predictions)
print("Confusion Matrix:")
print(conf_matrix)

# Calculate accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Accuracy:", accuracy))

# Calculate precision
precision <- conf_matrix[2, 2] / sum(binary_predictions)
print(paste("Precision:", precision))

# Calculate recall (sensitivity)
recall <- conf_matrix[2, 2] / sum(test_data$churn)
print(paste("Recall (Sensitivity):", recall))
```

Model Training and Evaluation

Data Splitting: Split the dataset into training (60%) and testing (40%) sets for model training and evaluation.

Gradient Boosting Model (GBM): Utilized GBM algorithm for predictive modeling due to its robustness and efficiency.

Model Evaluation: Evaluated the model's performance using confusion matrix, accuracy, precision, recall, and F1-score metrics.



Performance Visualization

ROC Curve Analysis: Plotted Receiver Operating Characteristic (ROC) curve to visualize the trade-off between true positive rate and false positive rate.

Area Under the Curve (AUC): Calculated AUC value to quantify the model's discriminatory power and effectiveness.

Deployment and Future Steps

```
#################### Testing Unseen Data ####################

# Load the test data
test_data_new <- read.csv("test.csv")
```

```
# Make predictions on the test set
predictions_2 <- predict(gbm_model, newdata = test_data_new, n.trees = 100, type = "response")
```

Testing on Unseen Data: Applied the trained model on new, unseen data for prediction using the same feature engineering techniques.

Submission: Generated predictions for customer churn on the test dataset and prepared a submission file in CSV format for deployment.

Future Steps: Recommendations for further enhancements such as hyperparameter tuning, feature selection, and exploring alternative algorithms for improved model performance.

Conclusion

This executive summary provides a concise overview of the code's workflow, emphasizing data processing, modeling, evaluation, and deployment aspects for predicting customer churn for a large online retailer.