MKTG 562: Group Assignment 2

Aishwary Jadhav, Jaivardhan Chauhan, Maitreyi Ekbote, Shreyansh Bhatia

1) Running a logistic regression of buyer variable on all other variables in the dataset gives the below result:

```
> summary(logit_model)

Call:
glm(formula = buyer ~ ., family = binomial, data = data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -1.555e+00  1.389e-01 -11.196  < 2e-16 ***
customer_id             -3.628e-06  3.449e-06  -1.052  0.29291
customer_typerestaurant -8.158e-01  4.035e-02 -20.218  < 2e-16 ***
last_purch              -9.458e-02  3.879e-03 -24.381  < 2e-16 ***
dollars                  2.608e-04  3.707e-04   0.703  0.48181
beverage1               -3.071e-01  2.688e-01  -1.143  0.25319
beverage2                3.783e-01  2.678e-01   1.413  0.15778
beverage3               -7.157e-01  2.691e-01  -2.659  0.00784 **
beverage4               -3.693e-01  2.677e-01  -1.380  0.16767
beverage5                8.552e-01  2.681e-01   3.190  0.00142 **
beverage6                4.692e-02  2.685e-01   0.175  0.86131
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 23817  on 39999  degrees of freedom
Residual deviance: 19373  on 39989  degrees of freedom
AIC: 19395

Number of Fisher Scoring iterations: 6
```

The logistic regression results reveal factors influencing the likelihood of customers purchasing the new beverage. Restaurants are significantly less likely to buy compared to bars, with an odds ratio of 0.442, suggesting that bars should be the primary target for future marketing efforts. Also, the recency of the last purchase strongly impacts buying behavior, with each additional week since the last purchase reducing the likelihood of ordering the new product. This highlights the importance of targeting recent buyers in marketing campaigns, as they are far more likely to engage with the promotion.

After calculating odds ratio, we get the below result:

	Variable	OR	Lower_CI	Upper_CI	P_Value
beverage5	beverage5	2.3517775	1.3905364	3.9777101	1.423678e-03
beverage2	beverage2	1.4597784	0.8635167	2.4672708	1.577811e-01
beverage6	beverage6	1.0480345	0.6189713	1.7736937	8.613076e-01
dollars	dollars	1.0002608	0.9995344	1.0009881	4.818111e-01
customer_id	customer_id	0.9999964	0.9999896	1.0000031	2.929070e-01
last_purch	last_purch	0.9097552	0.9028317	0.9166665	2.751867e-131
beverage1	beverage1	0.7355427	0.4341415	1.2453462	2.531878e-01
beverage4	beverage4	0.6912034	0.4088614	1.1676571	1.676710e-01
beverage3	beverage3	0.4888556	0.2883081	0.8281315	7.835179e-03
customer_typerestaurant	customer_typerestaurant	0.4423066	0.4086750	0.4787088	6.805758e-91
(Intercept)	(Intercept)	0.2111065	0.1607022	0.2770380	4.266591e-29

Below are the coefficients obtained for beverage 1, to beverage 6:

	Variable	OR	Lower_CI	Upper_CI	P_Value
beverage5	beverage5	2.3517775	1.3905364	3.9777101	0.001423678
beverage2	beverage2	1.4597784	0.8635167	2.4672708	0.157781134
beverage6	beverage6	1.0480345	0.6189713	1.7736937	0.861307567
beverage1	beverage1	0.7355427	0.4341415	1.2453462	0.253187836
beverage4	beverage4	0.6912034	0.4088614	1.1676571	0.167670993
beverage3	beverage3	0.4888556	0.2883081	0.8281315	0.007835179

Among the existing beverages, Beverage5 customers are over twice as likely to purchase the new beverage (OR = 2.35, p = 0.00142), indicating strong behavioral similarity between these products. In contrast, Beverage3 shows a negative effect (OR = 0.49, p = 0.00784), suggesting that its customers are less likely to adopt the new product. This suggests that the new beverage may appeal more to customers who already enjoy Beverage5, while customers of Beverage3 may have different preferences that do not align with the new offering.

From a marketing standpoint, the company should focus on bars, target recent buyers, and leverage Beverage5 customers for cross-promotions. Since certain beverage customers are more inclined to buy the new product, personalized marketing strategies should be employed to maximize conversions. Customers of Beverage3, however, may require additional incentives, different messaging, or even an alternative product offering to drive engagement. By refining its campaign strategy using these insights, the company can maximize profitability and avoid unnecessary sampling costs.

R Code:

```
logit_model <- glm(buyer ~ ., data = data, family = binomial)
summary(logit_model)
data$predicted_prob <- predict(logit_model, data = data, type = "response")

odds_ratios <- exp(coef(logit_model))
conf_int <- exp(confint(logit_model))


odds_ratios_df <- data.frame(
```

```
  Variable = names(odds_ratios),
  OR = odds_ratios,
  Lower_CI = conf_int[,1],
  Upper_CI = conf_int[,2],
  P_Value = summary(logit_model)$coefficients[,4]
)
odds_ratios_df <- odds_ratios_df %>% arrange(desc(OR))
print(odds_ratios_df)
beverage_vars <- odds_ratios_df %>% filter(grepl("beverage", Variable))
print(beverage_vars)
```

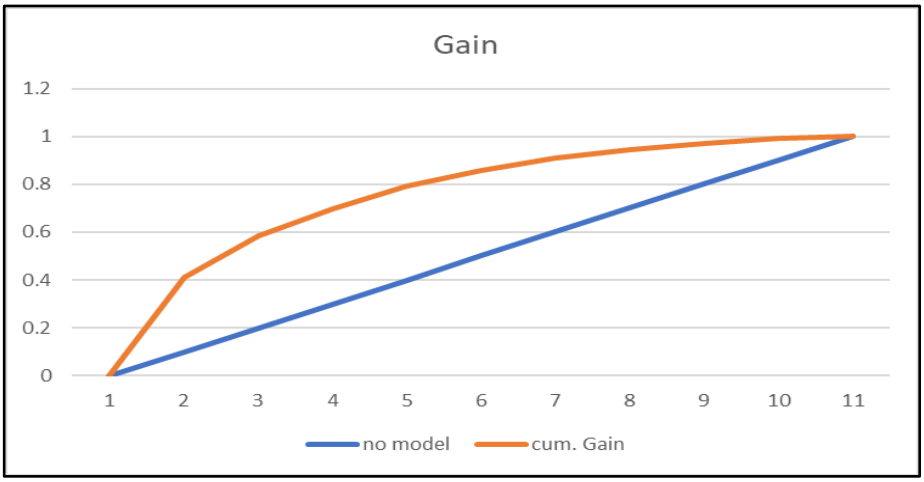
2) Using the predicted probabilities for creating deciles gives the below dataframe with number of customers and buyers:

	decile	count	buyers	avg_response_rate
1	1	4000	1448	0.36200
2	2	4000	608	0.15200
3	3	4000	399	0.09975
4	4	4000	330	0.08250
5	5	4000	229	0.05725
6	6	4000	179	0.04475
7	7	4000	127	0.03175
8	8	4000	86	0.02150
9	9	4000	78	0.01950
10	10	4000	33	0.00825

The Gain table in Excel looks like the following:

no model	decile	number of customers	number of respondents	cummulative number of customers	cummulative number of respondents	gain	cum. Gain
0	0	0	0	0	0	0	0
0.1	1	4000	1448	4000	1448	0.4117	0.411715
0.2	2	4000	608	8000	2056	0.1729	0.584589
0.3	3	4000	399	12000	2455	0.1134	0.698038
0.4	4	4000	330	16000	2785	0.0938	0.791868
0.5	5	4000	229	20000	3014	0.0651	0.85698
0.6	6	4000	179	24000	3193	0.0509	0.907876
0.7	7	4000	127	28000	3320	0.0361	0.943986
0.8	8	4000	86	32000	3406	0.0245	0.968439
0.9	9	4000	78	36000	3484	0.0222	0.990617
1	10	4000	33	40000	3517	0.0094	1
			3517				

Below is the Gain Chart:



The gain chart and table indicate that the logistic regression model effectively ranks customers based on their likelihood to purchase. The top 10% of customers (Decile 1) account for 41.17% of total buyers, and the top 20% (Deciles 1-2) capture nearly 58.46% of buyers, confirming that a small subset of high-probability customers drives most sales. This suggests that instead of sending samples to all 40,000 customers, targeting the top deciles (e.g., top 30%) could significantly improve marketing efficiency, reducing costs while maximizing conversions. Conversely, the bottom deciles contribute minimally to purchases, indicating that sending samples to these customers would be cost-ineffective.

From a marketing perspective, the company should focus its free sample campaign on the top-performing deciles, leveraging personalized promotions and strategic targeting for high-potential buyers. Deciles 1-3 (top 30%) should be the primary focus to optimize marketing spend, while lower-ranked deciles should receive alternative engagement strategies (e.g., email marketing instead of free samples). Additionally, cross-selling to Beverage5 customers (who have a strong correlation with new product purchases) could further enhance sales. This data-driven approach ensures higher ROI and reduced wastage in promotional efforts.

R Code:

```
data <- data %>% mutate(decile = 11 - ntile(predicted_prob, 10))

decile_summary <- data %>%
  group_by(decile) %>%
  summarise(
    count = n(),
    buyers = sum(as.numeric(buyer) == 1),
    avg_response_rate = mean(as.numeric(buyer))
  )
```

3) The break-even rate is 0.05357 or 5.357%.

On calculating whether the response probability of the logistic regression quintiles is above or below the break-even rate, we get the following table:

	decile	count	buyers	avg_response_rate	above_break_even
1	1	4000	1448	0.36200	TRUE
2	2	4000	608	0.15200	TRUE
3	3	4000	399	0.09975	TRUE
4	4	4000	330	0.08250	TRUE
5	5	4000	229	0.05725	TRUE
6	6	4000	179	0.04475	FALSE
7	7	4000	127	0.03175	FALSE
8	8	4000	86	0.02150	FALSE
9	9	4000	78	0.01950	FALSE
10	10	4000	33	0.00825	FALSE

The total profit for 220000 customers comes out to be $5983210.

The break-even response rate is calculated at 5.36% (0.05357), meaning that a customer must have at least a 5.36% probability of purchasing for the free sample campaign to be profitable. Based on the decile summary, deciles 1-5 have response rates above the break-even threshold, while deciles 6-10 fall below it. This suggests that the top 50% of customers (deciles 1-5) should be targeted with the free sample campaign, while the bottom 50% (deciles 6-10) should not receive samples, as their low purchase probability would lead to financial losses. This targeted approach optimizes cost efficiency and maximizes profitability.

For the full customer base of 220,000 customers, the company should only send free samples to customers in the top five deciles, totaling approximately 110,000 customers. The expected number of purchases from these targeted customers is calculated based on weighted response rates, leading to total projected orders and an estimated total profit of $5,983,210. This confirms that strategic targeting of high-probability buyers significantly improves financial outcomes, preventing unnecessary expenses on low-converting customers and ensuring a high return on investment.

R Code:

```
sample_cost <- 30
profit_per_sale <- 560
total_customers <- 220000

break_even_rate <- sample_cost / profit_per_sale
```

```
decile_summary <- decile_summary %>%
  mutate(
    above_break_even = avg_response_rate > break_even_rate
  )

profitable_customers  <-  sum(decile_summary$count[decile_summary$above_break_even])  /
sum(decile_summary$count) * total_customers

expected_orders <- sum(decile_summary$count[decile_summary$above_break_even] *

decile_summary$avg_response_rate[decile_summary$above_break_even]) /
  sum(decile_summary$count[decile_summary$above_break_even]) * profitable_customers

total_profit <- expected_orders * profit_per_sale - (profitable_customers * sample_cost)
```

4) **R Code**:

```
thresholds <- seq(0, 1, 0.001)
best_thresh  <- 0
best_accuracy <- 0

for (t in thresholds) {
  predicted_class <- ifelse(data$predicted_prob >= t, 1, 0)
  acc <- mean(predicted_class == data$buyer)  # Classification accuracy
  if (acc > best_accuracy) {
    best_accuracy <- acc
    best_thresh <- t
  }
}

cat("Best Threshold = ", best_thresh, "\n")
cat("Max Accuracy = ", best_accuracy, "\n")
```

Best Threshold = 0.527
Max Accuracy = 0.9179

```
predicted_class <- ifelse(data$predicted_prob >= best_thresh, 1, 0)
library(caret)
conf_mat <- confusionMatrix(
  factor(predicted_class, levels = c(0,1)),
  factor(data$buyer,         levels = c(0,1))
)

Conf_mat
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 36279  3080
         1   204   437

               Accuracy : 0.9179
                 95% CI : (0.9152, 0.9206)
    No Information Rate : 0.9121
    P-Value [Acc > NIR] : 1.701e-05

                  Kappa : 0.1882

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9944
            Specificity : 0.1243
         Pos Pred Value : 0.9217
         Neg Pred Value : 0.6817
             Prevalence : 0.9121
         Detection Rate : 0.9070
   Detection Prevalence : 0.9840
      Balanced Accuracy : 0.5593

       'Positive' Class : 0
```

Now, computing the profit for the 220,000 customers using the confusion matrix from the 40,000-customer sample. We have:

True Positives (TP) = 437 (predicted = 1 and actual = 1)

Predicted Positives (PP) = 641 (sum of predicted = 1)

From the problem statement:

Profit per actual buyer = $560 ($710 revenue − $110 production − $40 shipping).

Cost per sample sent = $30 ($11 product + $19 shipping).

We found the optimal threshold to be 0.527 by maximizing the classification accuracy, which reached about 91.79%. At that threshold, there were 437 true positives and 641 predicted positives in the 40,000-customer test sample. From these numbers, the sample profit is (437 x $560) - (641 x $30) = $225,490. Scaling up to the 220,000-customer population (5.5 times the sample size) yields an estimated profit of approximately $1.24 million. This threshold-based strategy focuses on reducing misclassifications rather than directly accounting for profit, so it often targets fewer customers compared to a break-even approach.

5) In Question 3, the break-even based targeting strategy, which sends samples to anyone whose predicted probability exceeds the cost-to-profit ratio resulted in a total profit of about $5.98 million. In contrast, the threshold-based approach from Q4 resulted in a total profit of about $1.24 million when applied to the entire customer base. The reason for this difference is that the break-even rule explicitly targets customers whose expected gain exceeds the cost of sending a sample, while the threshold approach aims to maximize classification accuracy without considering the relative monetary impact of false positives and false negatives. Consequently, focusing on break-even probability tends to generate higher overall profit, whereas focusing on classification accuracy can lead to lower profitability.

6) Average Accuracy score of the model: 91.78%

R Code:

```
set.seed(123)


# Create 5 folds based on the buyer variable
folds <- createFolds(data$buyer, k = 5)
accuracy_scores <- numeric(length(folds))


for (i in seq_along(folds)) {
  # Define train and test splits
  test_indices <- folds[[i]]
  train_data <- data[-test_indices, ]
  test_data  <- data[test_indices, ]
```

```
  model_cv <- glm(buyer ~ ., data = train_data, family = binomial)

  pred_prob <- predict(model_cv, newdata = test_data, type =
"response")

  pred_class <- ifelse(pred_prob >= 0.529, 1, 0)

  # Calculate the accuracy for this fold
  accuracy_scores[i] <- mean(pred_class == test_data$buyer)
}

avg_accuracy <- mean(accuracy_scores)
cat("Average 5-Fold CV Accuracy:", avg_accuracy, "\n")
```

Accuracy (everyone is a buyer): 0.087925

Accuracy (no one is a buyer): 0.912075

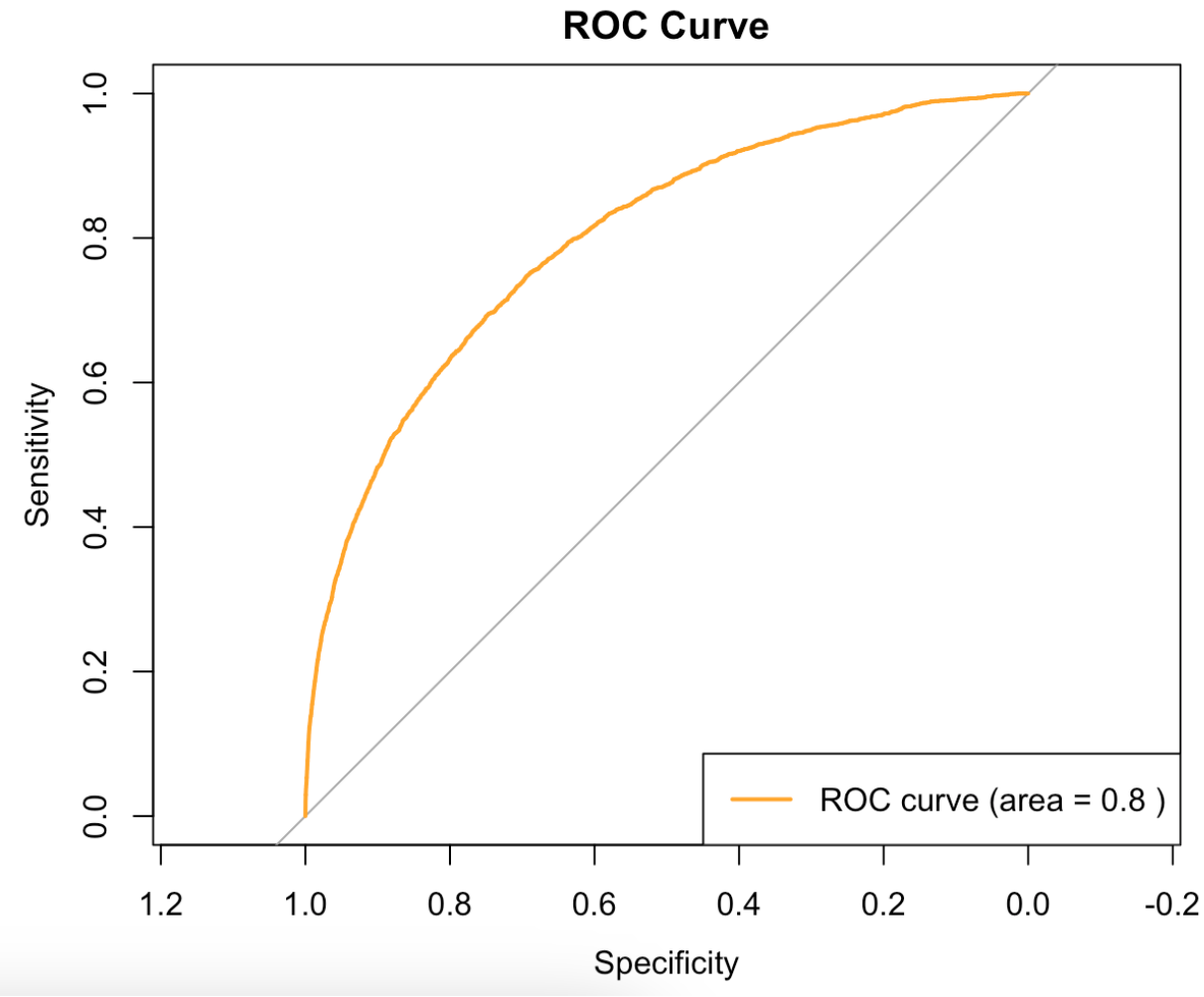
In summary, while the overall accuracy of your logistic model appears acceptable in terms of a number (around 91.8%), if the baseline “no one is a buyer” rule already achieves a similar accuracy, you might be seeing a counterintuitive situation where high accuracy does not necessarily translate into a model that identifies buyers well.

R Code:

```
ll_buyer_preds <- rep(1, nrow(data))
accuracy_all_buyer <- mean(all_buyer_preds == data$buyer)
cat("Accuracy (everyone is a buyer):", accuracy_all_buyer, "\n")
```

```
all_nonbuyer_preds <- rep(0, nrow(data))
accuracy_all_nonbuyer <- mean(all_nonbuyer_preds == data$buyer)
cat("Accuracy (no one is a buyer):", accuracy_all_nonbuyer, "\n")
```

7)



R Code:

```
Library(pROC)
roc_curve <- roc(data$buyer, data$predicted_prob)

# Plot the ROC curve
```

```
plot(roc_curve, col = "orange", lwd = 2, main = "ROC Curve")

auc_value <- auc(roc_curve)

legend("bottomright", legend = paste("ROC curve (area =", round(auc_value, 2), ")"), col = "orange", lwd = 2)
```