Task: Predicting Customer Purchase Behavior (Probability Estimation)

Scenario:

You work for an e-commerce company. You have customer data that includes whether each customer

clicked on a product link and whether they eventually made a purchase.

Your task is to estimate probabilities and answer questions like:

What is the probability that a customer makes a purchase given they clicked a link?

What is the overall probability of a purchase?

What is the probability that a customer did not purchase even after clicking?

# Load necessary libraries

library(dplyr)

library(ggplot2)

# Load the dataset

data <- read.csv("path\_to\_your\_dataset.csv")

# Filter relevant events

clicks <- data %>% filter(event\_type == "view")

purchases <- data %>% filter(event\_type == "purchase")

# Calculate total number of sessions

total\_sessions <- data %>% select(session\_id) %>% distinct() %>% nrow()

# Calculate number of sessions with at least one click

sessions\_with\_clicks <- clicks %>% select(session\_id) %>% distinct() %>% nrow()

# Calculate number of sessions with at least one purchase

sessions\_with\_purchases <- purchases %>% select(session\_id) %>% distinct() %>% nrow()

# Calculate number of sessions with both clicks and purchases

sessions\_with\_both <- clicks %>%

inner\_join(purchases, by = "session\_id") %>%

select(session\_id) %>%

distinct() %>%

nrow()

# Calculate probabilities

p\_purchase <- sessions\_with\_purchases / total\_sessions

p\_click <- sessions\_with\_clicks / total\_sessions

p\_purchase\_given\_click <- sessions\_with\_both / sessions\_with\_clicks

p\_no\_purchase\_given\_click <- 1 - p\_purchase\_given\_click

# Print probabilities

cat("P(Purchase):", p\_purchase, "\n")

cat("P(Click):", p\_click, "\n")

cat("P(Purchase | Click):", p\_purchase\_given\_click, "\n")

cat("P(No Purchase | Click):", p\_no\_purchase\_given\_click, "\n")

# Visualization

# Aggregate data for visualization

clicks\_purchases <- data %>%

filter(event\_type %in% c("view", "purchase")) %>%

group\_by(event\_type) %>%

summarise(count = n())

# Bar plot of Clicks vs. Purchases

ggplot(clicks\_purchases, aes(x = event\_type, y = count, fill = event\_type)) +

geom\_bar(stat = "identity") +

labs(x = "Event Type", y = "Count", title = "Number of Clicks and Purchases") +

scale\_fill\_manual(values = c("view" = "blue", "purchase" = "green")) +

theme\_minimal()

# We will simulate a real-world-like dataset for now (or use a real one if you prefer downloading from sources like Kaggle). The dataset includes:

# customer\_id: Unique ID for each customer

# email\_opened: 1 if the customer opened the email, 0 otherwise

# // link\_clicked: 1 if the customer clicked a link, 0 otherwise

set.seed(123)

# Simulate 1000 email campaigns

n <- 1000

data <- data.frame(

customer\_id = 1:n,

email\_opened = rbinom(n, 1, 0.6), # 60% open rate

link\_clicked = 0

)

# Assign link\_clicked only to those who opened the email (30% of openers)

data$link\_clicked[data$email\_opened == 1] <- rbinom(sum(data$email\_opened == 1), 1, 0.3)

head(data)

#Calculate Probabilities

# Total customers

total\_customers <- nrow(data)

# Customers who opened email

opened\_email <- sum(data$email\_opened == 1)

# Customers who clicked link

clicked\_link <- sum(data$link\_clicked == 1)

# Customers who opened AND clicked

opened\_and\_clicked <- sum(data$email\_opened == 1 & data$link\_clicked == 1)

# Customers who opened but DID NOT click

opened\_no\_click <- sum(data$email\_opened == 1 & data$link\_clicked == 0)

# Probabilities

p\_click\_given\_open <- opened\_and\_clicked / opened\_email

p\_click <- clicked\_link / total\_customers

p\_no\_click\_given\_open <- opened\_no\_click / opened\_email

# Output results

cat("P(Click | Opened Email):", round(p\_click\_given\_open, 3), "\n")

cat("P(Click):", round(p\_click, 3), "\n")

cat("P(No Click | Opened Email):", round(p\_no\_click\_given\_open, 3), "\n")

#visualization

library(ggplot2)

ggplot(data, aes(x = factor(email\_opened), fill = factor(link\_clicked))) +

geom\_bar(position = "fill") +

labs(title = "Link Click Probability Given Email Opened",

x = "Email Opened (1 = Yes, 0 = No)",

y = "Proportion",

fill = "Link Clicked") +

scale\_fill\_manual(values = c("0" = "red", "1" = "green"))

Task: Rain Prediction Based on Cloud Cover

Scenario:

You work with a meteorological department. You have historical weather data for a city that includes information about cloud cover and rain occurrence.

Objective: Estimate Probabilities

What is the probability that it rains given it’s cloudy?

What is the overall probability of rain?

What is the probability that it does not rain even though it’s cloudy?

day: Day number

is\_cloudy: 1 if the day was cloudy, 0 otherwise

is\_rainy: 1 if it rained on that day, 0 otherwise

set.seed(100)

# Simulate 365 days

n <- 365

weather <- data.frame(

day = 1:n,

is\_cloudy = rbinom(n, 1, 0.5), # 50% of days are cloudy

is\_rainy = 0

)

# Assign rain conditionally - 40% chance of rain if cloudy, 10% if not

weather$is\_rainy[weather$is\_cloudy == 1] <- rbinom(sum(weather$is\_cloudy == 1), 1, 0.4)

weather$is\_rainy[weather$is\_cloudy == 0] <- rbinom(sum(weather$is\_cloudy == 0), 1, 0.1)

head(weather)

#Calculate probabilities

# Total days

total\_days <- nrow(weather)

# Cloudy days

cloudy\_days <- sum(weather$is\_cloudy == 1)

# Rainy days

rainy\_days <- sum(weather$is\_rainy == 1)

# Cloudy and rainy

cloudy\_and\_rainy <- sum(weather$is\_cloudy == 1 & weather$is\_rainy == 1)

# Cloudy but not rainy

cloudy\_no\_rain <- sum(weather$is\_cloudy == 1 & weather$is\_rainy == 0)

# Calculating probabilities

p\_rain <- rainy\_days / total\_days

p\_rain\_given\_cloudy <- cloudy\_and\_rainy / cloudy\_days

p\_no\_rain\_given\_cloudy <- cloudy\_no\_rain / cloudy\_days

# Output

cat("P(Rain):", round(p\_rain, 3), "\n")

cat("P(Rain | Cloudy):", round(p\_rain\_given\_cloudy, 3), "\n")

cat("P(No Rain | Cloudy):", round(p\_no\_rain\_given\_cloudy, 3), "\n")

#visualization

library(ggplot2)

ggplot(weather, aes(x = factor(is\_cloudy), fill = factor(is\_rainy))) +

geom\_bar(position = "fill") +

labs(title = "Rain Given Cloudy Conditions",

x = "Is Cloudy (1 = Yes, 0 = No)",

y = "Proportion of Days",

fill = "Is Rainy") +

scale\_fill\_manual(values = c("0" = "skyblue", "1" = "navy")) +

theme\_minimal()

# 4. Health Diagnosis Prediction

# Real-World Health Data: Predict if a patient has a disease based on symptoms

# Goal:

# Estimate:

# P(Disease | Symptom)

# P(Disease)

# P(No Disease | Symptom)

set.seed(123)

# Simulate 1000 patients

n <- 1000

health <- data.frame(

patient\_id = 1:n,

has\_symptom = rbinom(n, 1, 0.5), # 50% have the symptom

has\_disease = 0

)

# 40% chance of disease if they have the symptom, 5% if they don't

health$has\_disease[health$has\_symptom == 1] <- rbinom(sum(health$has\_symptom == 1), 1, 0.4)

health$has\_disease[health$has\_symptom == 0] <- rbinom(sum(health$has\_symptom == 0), 1, 0.05)

# Probabilities

symptom <- sum(health$has\_symptom == 1)

disease <- sum(health$has\_disease == 1)

symptom\_and\_disease <- sum(health$has\_symptom == 1 & health$has\_disease == 1)

symptom\_no\_disease <- sum(health$has\_symptom == 1 & health$has\_disease == 0)

p\_disease <- disease / n

p\_disease\_given\_symptom <- symptom\_and\_disease / symptom

p\_no\_disease\_given\_symptom <- symptom\_no\_disease / symptom

cat("P(Disease):", round(p\_disease, 3), "\n")

cat("P(Disease | Symptom):", round(p\_disease\_given\_symptom, 3), "\n")

cat("P(No Disease | Symptom):", round(p\_no\_disease\_given\_symptom, 3), "\n")

# Add readable labels

health$Symptom <- ifelse(health$has\_symptom == 1, "Has Symptom", "No Symptom")

health$Disease <- ifelse(health$has\_disease == 1, "Has Disease", "No Disease")

# Proportional bar plot

ggplot(health, aes(x = Symptom, fill = Disease)) +

geom\_bar(position = "fill") +

scale\_y\_continuous(labels = scales::percent\_format()) +

labs(

title = "Probability of Disease Based on Symptom Presence",

x = "Symptom Status",

y = "Percentage",

fill = "Disease Status"

) +

scale\_fill\_manual(values = c("Has Disease" = "#e74c3c", "No Disease" = "#2ecc71")) +

theme\_minimal(base\_size = 14)

# Student Exam Performance Prediction

# Study hours vs. passing exam

set.seed(101)

# Simulate 800 students

n <- 800

students <- data.frame(

student\_id = 1:n,

studied = rbinom(n, 1, 0.7), # 70% studied

passed = 0

)

# If studied: 85% pass chance, else 30%

students$passed[students$studied == 1] <- rbinom(sum(students$studied == 1), 1, 0.85)

students$passed[students$studied == 0] <- rbinom(sum(students$studied == 0), 1, 0.3)

# Calculating probabilities

studied <- sum(students$studied == 1)

passed <- sum(students$passed == 1)

studied\_and\_passed <- sum(students$studied == 1 & students$passed == 1)

studied\_failed <- sum(students$studied == 1 & students$passed == 0)

p\_pass <- passed / n

p\_pass\_given\_studied <- studied\_and\_passed / studied

p\_fail\_given\_studied <- studied\_failed / studied

cat("P(Pass):", round(p\_pass, 3), "\n")

cat("P(Pass | Studied):", round(p\_pass\_given\_studied, 3), "\n")

cat("P(Fail | Studied):", round(p\_fail\_given\_studied, 3), "\n")

install.packages("ggplot2") # Run this if you haven't already

library(ggplot2)

# Convert binary to labels for better plot readability

students$Studied <- ifelse(students$studied == 1, "Studied", "Did Not Study")

students$Result <- ifelse(students$passed == 1, "Passed", "Failed")

# Plot: Proportion of Pass/Fail by Study Status

ggplot(students, aes(x = Studied, fill = Result)) +

geom\_bar(position = "fill") +

scale\_y\_continuous(labels = scales::percent\_format()) +

labs(

title = "Student Exam Results Based on Study Status",

x = "Study Status",

y = "Percentage",

fill = "Exam Result"

) +

scale\_fill\_manual(values = c("Passed" = "#1f77b4", "Failed" = "#ff7f0e")) +

theme\_minimal()