Customer_Churn_shiny_app

Bibek Shah

2024-10-19

```
# Load necessary libraries
library(shiny)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(fmsb)

# Read the data
data <- read.csv("C:/Users/Public/Documents/sem_4_R/Dataset/OneDrive_2024-09-05/customer churn/Telecom_

# Data preparation
data <- data %>% mutate_if(is.numeric, ~ ifelse(is.na(.), median(., na.rm = TRUE), .))
data$churn <- as.factor(data$churn)

# UI
ui <- fluidPage(
  titlePanel("Customer Churn Analysis"),
```

```
  sidebarLayout(
    sidebarPanel(
      selectInput("plotType", "Choose a Plot", choices = c(
        "Churn Distribution",
        "Total Revenue vs Churn",
        "Churn Rate by Area",
        "Density Plot of Monthly Revenue by Churn",
        "Churn vs. Mean Outgoing Usage",
        "Correlation Matrix",
        "Total Revenue by Churn Status",
        "Churn Rate Over Time",
        "Pairwise Comparison of Selected Features",
        "Churn Rate by Marital Status",
        "Distribution of Monthly Revenue",
        "Data Usage Distribution by Churn Status",
        "Total Revenue vs. Mean Outgoing Usage",
        "Outgoing Calls by Churn Status",
        "Feature Distributions Radar Chart"
      ))
    ),
    mainPanel(
      plotOutput("churnPlot")
    )
  )
)

# Server
server <- function(input, output) {
  output$churnPlot <- renderPlot({
    switch(input$plotType,
           "Churn Distribution" = {
             ggplot(data, aes(x = churn)) +
               geom_bar() +
               labs(title = "Churn Distribution", x = "Churn", y = "Count")
           },
           "Total Revenue vs Churn" = {
             ggplot(data, aes(x = totrev, fill = churn)) +
               geom_histogram(bins = 30) +
               labs(title = "Total Revenue vs Churn", x = "Total Revenue", y = "Count")
           },
           "Churn Rate by Area" = {
             ggplot(data, aes(x = area, fill = churn)) +
               geom_bar(position = "fill") +
               labs(title = "Churn Rate by Area", x = "Area", y = "Proportion") +
               scale_y_continuous(labels = scales::percent) +
               theme(axis.text.x = element_text(angle = 45, hjust = 1))
           },
           "Density Plot of Monthly Revenue by Churn" = {
             ggplot(data, aes(x = totmrc_Mean, fill = churn)) +
               geom_density(alpha = 0.5) +
               labs(title = "Density Plot of Monthly Revenue by Churn", x = "Monthly Revenue", y = "Dens
               theme_minimal()
           },
```

```
    "Churn vs. Mean Outgoing Usage" = {
      ggplot(data, aes(x = mou_Mean, y = churn, color = churn)) +
        geom_jitter(alpha = 0.5) +
        labs(title = "Churn vs. Mean Outgoing Usage", x = "Mean Outgoing Usage (MOU)", y = "Churn
        theme_minimal()
    },
    "Correlation Matrix" = {
      cor_matrix <- cor(data %>% select_if(is.numeric))
      corrplot(cor_matrix, method = "circle", type = "upper", order = "hclust",
               tl.col = "black", tl.srt = 45, title = "Correlation Matrix")
    },
    "Total Revenue by Churn Status" = {
      ggplot(data, aes(x = churn, y = totrev)) +
        geom_boxplot() +
        labs(title = "Total Revenue by Churn Status", x = "Churn Status", y = "Total Revenue") +
        theme_minimal()
    },
    "Churn Rate Over Time" = {
      ggplot(data, aes(x = months, fill = churn)) +
        geom_bar(position = "fill") +
        labs(title = "Churn Rate Over Time", x = "Months", y = "Proportion") +
        scale_y_continuous(labels = scales::percent) +
        theme_minimal()
    },
    "Pairwise Comparison of Selected Features" = {
      ggpairs(data %>% select(totrev, mou_Mean, totmrc_Mean, churn),
              mapping = aes(color = churn)) +
        labs(title = "Pairwise Comparison of Selected Features")
    },
    "Churn Rate by Marital Status" = {
      ggplot(data, aes(x = marital, fill = churn)) +
        geom_bar(position = "fill") +
        labs(title = "Churn Rate by Marital Status", x = "Marital Status", y = "Proportion") +
        scale_y_continuous(labels = scales::percent) +
        theme_minimal()
    },
    "Distribution of Monthly Revenue" = {
      ggplot(data, aes(x = totmrc_Mean, fill = churn)) +
        geom_histogram(bins = 30, alpha = 0.7, position = "identity") +
        labs(title = "Distribution of Monthly Revenue", x = "Monthly Revenue", y = "Count") +
        theme_minimal()
    },
    "Data Usage Distribution by Churn Status" = {
      ggplot(data, aes(x = churn, y = datovr_Mean, fill = churn)) +
        geom_violin(trim = FALSE) +
        labs(title = "Data Usage Distribution by Churn Status", x = "Churn Status", y = "Data Ove
        theme_minimal()
    },
    "Total Revenue vs. Mean Outgoing Usage" = {
      ggplot(data, aes(x = totrev, y = mou_Mean, color = churn)) +
        geom_point(alpha = 0.6) +
        labs(title = "Total Revenue vs. Mean Outgoing Usage", x = "Total Revenue", y = "Mean Outg
        theme_minimal() +
```

```
            geom_smooth(method = "lm", se = FALSE)
        },
        "Outgoing Calls by Churn Status" = {
          ggplot(data, aes(x = churn, y = totcalls)) +
            geom_boxplot() +
            labs(title = "Outgoing Calls by Churn Status", x = "Churn Status", y = "Total Calls") +
            theme_minimal()
        },
        "Feature Distributions Radar Chart" = {
          feature_data <- data %>%
            group_by(churn) %>%
            summarise(across(c(totmrc_Mean, mou_Mean, totrev, datovr_Mean), mean))

          feature_data <- rbind(rep(200, 4), rep(0, 4), feature_data)
          colnames(feature_data) <- c("totmrc_Mean", "mou_Mean", "totrev", "datovr_Mean")
          rownames(feature_data) <- c("Max", "Min", "Churned", "Not Churned")

          radarchart(feature_data, axistype = 1, pcol = c("red", "blue"), plwd = 2, plty = 1)
        }
    )
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```