fashion retail sales machine learning project3

Ayobamidele

2025-05-30

# Load required libraries

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.3

##   
## 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(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.3

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.4.3

##   
## Attaching package: 'lubridate'

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

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.4.3

library(caret)

## Warning: package 'caret' was built under R version 4.4.3

## Loading required package: lattice

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.4.3

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:dplyr':  
##   
## combine

library(e1071)

## Warning: package 'e1071' was built under R version 4.4.3

library(forecast)

## Warning: package 'forecast' was built under R version 4.4.3

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(cluster)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.4.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(dplyr)  
library(lubridate)

# Load data

data <- read.csv("Fashion\_Retail\_Sales.csv", stringsAsFactors = FALSE)  
View(data)  
str(data)

## 'data.frame': 3400 obs. of 6 variables:  
## $ Customer.Reference.ID: int 4018 4115 4019 4097 3997 4080 4055 3973 4044 4010 ...  
## $ Item.Purchased : chr "Handbag" "Tunic" "Tank Top" "Leggings" ...  
## $ Purchase.Amount..USD.: int 4619 2456 2102 3126 3003 2914 2571 2419 4771 4233 ...  
## $ Date.Purchase : chr "05-02-2023" "11-07-2023" "23-03-2023" "15-03-2023" ...  
## $ Review.Rating : num NA 2 4.1 3.2 4.7 4.5 1.3 4.6 4.1 NA ...  
## $ Payment.Method : chr "Credit Card" "Credit Card" "Cash" "Cash" ...

#Checking Missing Values

sapply(data, function(x) sum(is.na(x)))

## Customer.Reference.ID Item.Purchased Purchase.Amount..USD.   
## 0 0 650   
## Date.Purchase Review.Rating Payment.Method   
## 0 324 0

#Removing Missing Values

data<-na.omit(data)  
str(data)

## 'data.frame': 2487 obs. of 6 variables:  
## $ Customer.Reference.ID: int 4115 4019 4097 3997 4080 4055 3973 4044 4108 4067 ...  
## $ Item.Purchased : chr "Tunic" "Tank Top" "Leggings" "Wallet" ...  
## $ Purchase.Amount..USD.: int 2456 2102 3126 3003 2914 2571 2419 4771 2356 4418 ...  
## $ Date.Purchase : chr "11-07-2023" "23-03-2023" "15-03-2023" "27-11-2022" ...  
## $ Review.Rating : num 2 4.1 3.2 4.7 4.5 1.3 4.6 4.1 4.8 3.4 ...  
## $ Payment.Method : chr "Credit Card" "Cash" "Cash" "Cash" ...  
## - attr(\*, "na.action")= 'omit' Named int [1:913] 1 10 13 16 20 21 36 38 40 48 ...  
## ..- attr(\*, "names")= chr [1:913] "1" "10" "13" "16" ...

# Assign correct column names manually

colnames(data) <- c("CustomerID", "ItemPurchased", "PurchaseAmountUSD",   
 "DatePurchase", "ReviewRating", "PaymentMethod")

# Convert DatePurchase to proper date format

data$DatePurchase <- as.Date(data$DatePurchase, format = "%d-%m-%Y")  
View(data)

# Extract Month and Day of Week

data$Month <- months(data$DatePurchase)  
data$DayOfWeek <- weekdays(data$DatePurchase)

# Train/test split

set.seed(123)  
train\_index <- createDataPartition(data$ReviewRating, p = 0.8, list = FALSE)  
train\_reg <- data[train\_index, ]  
test\_reg <- data[-train\_index, ]

# Train Random Forest

rf\_reg\_model <- randomForest(  
 ReviewRating ~ ItemPurchased + PurchaseAmountUSD + Month + DayOfWeek,  
 data = train\_reg,  
 ntree = 100,  
 importance = TRUE  
)

#predict and evaluate

predictions\_reg <- predict(rf\_reg\_model, newdata = test\_reg)  
postResample(predictions\_reg, test\_reg$ReviewRating)

## RMSE Rsquared MAE   
## 1.147445295 0.001918921 0.987587874

test\_reg$Predicted\_rating <- predictions\_reg   
View(test\_reg)

# Load libraries  
library(tsibble)

## Warning: package 'tsibble' was built under R version 4.4.3

## Registered S3 method overwritten by 'tsibble':  
## method from   
## as\_tibble.grouped\_df dplyr

##   
## Attaching package: 'tsibble'

## The following object is masked from 'package:lubridate':  
##   
## interval

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

library(forecast)  
library(ggplot2)  
# Load dataset (update path if necessary)  
data <- read.csv("Fashion\_Retail\_Sales.csv")  
# Assign correct column names manually  
colnames(data) <- c("CustomerID", "ItemPurchased", "PurchaseAmountUSD",   
 "DatePurchase", "ReviewRating", "PaymentMethod")  
  
# Convert numeric columns  
data$PurchaseAmountUSD <- as.numeric(data$PurchaseAmountUSD)  
data$ReviewRating <- as.numeric(data$ReviewRating)  
  
# Convert date to proper format  
data$DatePurchase <- as.Date(data$DatePurchase, format = "%d-%m-%Y")  
# Load libraries  
library(dplyr)  
  
# Make sure PurchaseAmountUSD is numeric  
data$PurchaseAmountUSD <- as.numeric(data$PurchaseAmountUSD)  
  
# Aggregate daily sales  
daily\_sales <- data %>%  
 group\_by(DatePurchase) %>%  
 summarise(TotalSales = sum(PurchaseAmountUSD, na.rm = TRUE))  
str(daily\_sales)

## tibble [365 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ DatePurchase: Date[1:365], format: "2022-10-02" "2022-10-03" ...  
## $ TotalSales : num [1:365] 822 929 1071 1160 411 ...

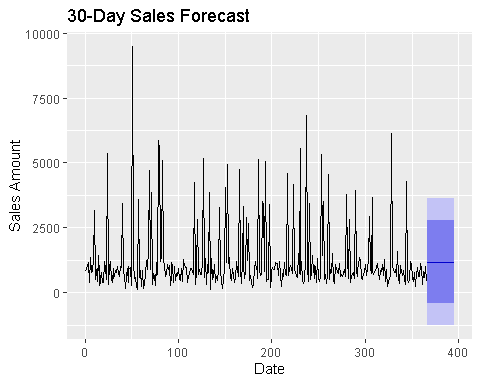
# Fit ARIMA model  
fit\_arima <- auto.arima(daily\_sales$TotalSales)  
fit\_arima

## Series: daily\_sales$TotalSales   
## ARIMA(0,0,0) with non-zero mean   
##   
## Coefficients:  
## mean  
## 1180.6904  
## s.e. 64.9489  
##   
## sigma^2 = 1543931: log likelihood = -3118.01  
## AIC=6240.02 AICc=6240.05 BIC=6247.82

# Forecast next 30 days  
forecast\_sales <- forecast(fit\_arima, h = 30)  
forecast\_sales

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 366 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 367 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 368 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 369 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 370 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 371 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 372 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 373 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 374 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 375 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 376 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 377 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 378 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 379 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 380 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 381 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 382 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 383 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 384 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 385 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 386 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 387 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 388 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 389 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 390 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 391 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 392 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 393 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 394 1180.69 -411.7019 2773.083 -1254.663 3616.044  
## 395 1180.69 -411.7019 2773.083 -1254.663 3616.044

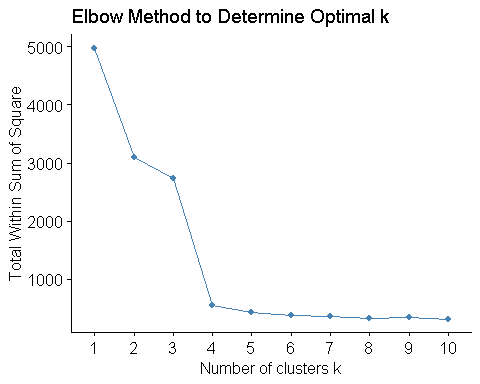
# Plot forecast  
autoplot(forecast\_sales) +  
 ggtitle("30-Day Sales Forecast") +  
 xlab("Date") +  
 ylab("Sales Amount")



# Load necessary libraries  
library(dplyr)  
library(cluster)  
library(factoextra)  
  
# Step 1: Load the dataset  
data <- read.csv("Fashion\_Retail\_Sales.csv")  
  
  
# Step 2: Assign correct column names manually  
colnames(data) <- c("CustomerID", "ItemPurchased", "PurchaseAmountUSD",   
 "DatePurchase", "ReviewRating", "PaymentMethod")  
  
# Step 3: Convert numeric columns  
data$PurchaseAmountUSD <- as.numeric(data$PurchaseAmountUSD)  
data$ReviewRating <- as.numeric(data$ReviewRating)  
  
# Step 4: Remove rows with missing values in relevant columns  
cluster\_data <- data %>%  
 select(PurchaseAmountUSD, ReviewRating) %>%  
 na.omit()  
str(cluster\_data)

## 'data.frame': 2487 obs. of 2 variables:  
## $ PurchaseAmountUSD: num 2456 2102 3126 3003 2914 ...  
## $ ReviewRating : num 2 4.1 3.2 4.7 4.5 1.3 4.6 4.1 4.8 3.4 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:913] 1 10 13 16 20 21 36 38 40 48 ...  
## ..- attr(\*, "names")= chr [1:913] "1" "10" "13" "16" ...

# Step 5: Scale data  
cluster\_scaled <- scale(cluster\_data)  
  
# Step 6: Determine optimal number of clusters (k)  
fviz\_nbclust(cluster\_scaled, kmeans, method = "wss") +  
 labs(title = "Elbow Method to Determine Optimal k")



# Step 7: Perform K-means clustering (with 3 clusters as example)  
kmeans\_model <- kmeans(cluster\_scaled, centers = 3, nstart = 25)  
  
# Step 8: Add cluster labels back to original data  
clustered\_data <- data[match(rownames(cluster\_data), rownames(data)), , drop = FALSE]  
head(clustered\_data)

## CustomerID ItemPurchased PurchaseAmountUSD DatePurchase ReviewRating  
## 2 4115 Tunic 2456 11-07-2023 2.0  
## 3 4019 Tank Top 2102 23-03-2023 4.1  
## 4 4097 Leggings 3126 15-03-2023 3.2  
## 5 3997 Wallet 3003 27-11-2022 4.7  
## 6 4080 Onesie 2914 11-12-2022 4.5  
## 7 4055 Jacket 2571 08-07-2023 1.3  
## PaymentMethod  
## 2 Credit Card  
## 3 Cash  
## 4 Cash  
## 5 Cash  
## 6 Credit Card  
## 7 Cash

clustered\_data$Cluster <- as.factor(kmeans\_model$cluster)  
  
# Step 9: Visualize clusters  
fviz\_cluster(kmeans\_model, data = cluster\_scaled,   
 palette = c("#E7B800", "#4E79A7", "#A0C4E8"),  
 main = "Clustering of Products by Purchase Amount & Review Rating")

