Assignment 3

Alex Khaykin

2024-11-24

library(tidyverse)  
library(lubridate)  
library(ggplot2)  
library(corrplot)  
library(e1071)  
library(caret)

# INTRO

The landscape of modern business increasingly relies on data-driven decision-making, and effective analysis of sales data can significantly enhance operational efficiency and customer satisfaction. For my project, I focus on understanding the relationship between two critical factors influencing Fulfilment\_Speed: Order.Priority and and other influencing factors such as geography and the companies Total.Revenue. By SVM model, I aim to uncover patterns within my chosen dataset composed of 10000 sales transactions, examining how these variables interact and impact overall sales outcomes.

Order priority typically reflects the urgency and importance assigned to each sale, potentially impacting the fulfillment process and customer experience. Fulfillment speed, on the other hand, denotes the time taken to process and deliver an order. In a competitive market, optimizing both order priority and fulfillment speed is essential for meeting customer expectations and driving sales growth. Through SVM, I will explore whether order priority status can predict fulfillment speed, providing valuable insights for stakeholders looking to enhance their operational strategies.

This project will articulate the statistical relationship between order priority and fulfillment speed and provide actionable recommendations for improving order management and fulfillment processes. By leveraging SVM analysis, businesses can make informed decisions to enhance their performance in a fast-paced market environment.

# DATA

Reading in the data downloaded from the site and converting it to a data frame from where I will be unitizing R to run summary statistics to learn about the data and the information contained within.

sales\_10000 <- read.csv("C:\\Users\\akhay\\OneDrive\\Documents\\DATA\_SCIENCE\\DATA\_622\\Assignments\\Assignment\_2\\10000 Sales Records.csv")

### Summary

This section gives the names of the columns(variable), and the basic statistics of the dataset. There seems to be no missing data in this dataset.

summary(sales\_10000)

## Region Country Item.Type Sales.Channel   
## Length:10000 Length:10000 Length:10000 Length:10000   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Order.Priority Order.Date Order.ID Ship.Date   
## Length:10000 Length:10000 Min. :100089156 Length:10000   
## Class :character Class :character 1st Qu.:321806669 Class :character   
## Mode :character Mode :character Median :548566305 Mode :character   
## Mean :549871874   
## 3rd Qu.:775998104   
## Max. :999934232   
## Units.Sold Unit.Price Unit.Cost Total.Revenue   
## Min. : 2 Min. : 9.33 Min. : 6.92 Min. : 168   
## 1st Qu.: 2531 1st Qu.:109.28 1st Qu.: 56.67 1st Qu.: 288551   
## Median : 4962 Median :205.70 Median :117.11 Median : 800051   
## Mean : 5003 Mean :268.14 Mean :188.81 Mean :1333355   
## 3rd Qu.: 7472 3rd Qu.:437.20 3rd Qu.:364.69 3rd Qu.:1819143   
## Max. :10000 Max. :668.27 Max. :524.96 Max. :6680027   
## Total.Cost Total.Profit   
## Min. : 125 Min. : 43.4   
## 1st Qu.: 164786 1st Qu.: 98329.1   
## Median : 481606 Median : 289099.0   
## Mean : 938266 Mean : 395089.3   
## 3rd Qu.:1183822 3rd Qu.: 566422.7   
## Max. :5241726 Max. :1738178.4

The dataset contains 10,000 sales records, with 14 columns. However for the purpose of the question I am asking, I will be treating these as company records and dropping the following columns: Country, Order Id. Additional y I will be making a new primary predictor variable Fullfilment Speed, which is the lag between Order Date and Ship Date, after which I will drop the two date columns entirely. I am treating this as the average fulfillment speed for each of the hypothetical companies and their records contained in the dataset.

sales\_10000 <- sales\_10000 %>%   
 select(-Country, -Order.ID) %>%   
 mutate(Order.Date=as.Date(Order.Date, format="%m/%d/%Y"),   
 Ship.Date=as.Date(Ship.Date, format= "%m/%d/%Y"),  
 Fulfilment\_Speed =as.numeric(Ship.Date - Order.Date)) %>%   
 select(-Order.Date, -Ship.Date) %>%   
 mutate\_if(is.character, as.factor)

# MODELS

### Order Priority

In an attempt to predict whether an order would come in as high priority, I first re-coded the original order priority variable as either high or not high (everything else). I then built another decision tree model that uses the following predictors. Region, Item Type, Sales Channel, Units Sold, Unit Price, Unit Cost Total Revenue, Total Cost, Total Profit, and Fulfillment Speed. I again used a partition of 75 / 25.

set.seed(25)  
sales\_10000 <- sales\_10000 %>%   
 mutate(Order.Priority=as.factor(ifelse(Order.Priority=="H","High", "Not High")))  
ind <- createDataPartition(sales\_10000$Order.Priority, p=0.75, list=FALSE)  
train\_1 <- sales\_10000[ind, ]  
test\_1 <- sales\_10000[-ind, ]

mod1 <- svm(Order.Priority ~., data = train\_1, kernel = "linear")  
mod2 <- svm(Order.Priority ~., data = train\_1, kernel = "radial")  
mod3 <- svm(Order.Priority ~., data = train\_1, kernel = "sigmoid")

preds\_1 <- predict(mod1, newdata = test\_1, type = "class")  
confusionMatrix(preds\_1, test\_1$Order.Priority)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction High Not High  
## High 0 0  
## Not High 625 1874  
##   
## Accuracy : 0.7499   
## 95% CI : (0.7324, 0.7668)  
## No Information Rate : 0.7499   
## P-Value [Acc > NIR] : 0.5107   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.7499   
## Prevalence : 0.2501   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : High   
##

preds\_2 <- predict(mod2, newdata = test\_1, type = "class")  
confusionMatrix(preds\_2, test\_1$Order.Priority)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction High Not High  
## High 0 0  
## Not High 625 1874  
##   
## Accuracy : 0.7499   
## 95% CI : (0.7324, 0.7668)  
## No Information Rate : 0.7499   
## P-Value [Acc > NIR] : 0.5107   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.7499   
## Prevalence : 0.2501   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : High   
##

preds\_3 <- predict(mod3, newdata = test\_1, type = "class")  
confusionMatrix(preds\_3, test\_1$Order.Priority)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction High Not High  
## High 82 260  
## Not High 543 1614  
##   
## Accuracy : 0.6787   
## 95% CI : (0.66, 0.697)  
## No Information Rate : 0.7499   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : -0.0089   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.13120   
## Specificity : 0.86126   
## Pos Pred Value : 0.23977   
## Neg Pred Value : 0.74826   
## Prevalence : 0.25010   
## Detection Rate : 0.03281   
## Detection Prevalence : 0.13685   
## Balanced Accuracy : 0.49623   
##   
## 'Positive' Class : High   
##

The confusion matrix for either linear or radial models had the highest accuracy at 0.75 and the sigmoid model had the lowest accuracy at 0.68. I is notable that the linear and radial models have the same accuracy as the random forest model from the previous assignment, as well as identical sensitivity and specificity. The specificity is 100%, meaning that the model can identify when an order is not a high priority, but has zero percent sensitivity, meaning that it is unable to predict when it is high priority. This is due to the data being artificial and not real sales data.

### Which algorithm is recommended to get more accurate results?

Higher accuracy results would depend on context and the data more than it would on the choice of which predictive model. In the first article we were assigned to read Ahmad in 2021, SVM was not compared to ensemble methods used to predict COVID19 infections. They found that random forest had the highest accuracy for predicting the classifier (Table 2 in the paper). The second case paper by Guhathakurata in 2021 used only SVM, and were able to successfully predict severe COVID19 cases with very high F1 score of 97%. This case demonstrates a time when SVM was a successful classifier, but no other models were tested and compared. In my data chosen data and for the question I am asking, both random forest and SVM are equal classifiers, yielding an accuracy of 75% but 0% sensitivity. Meaning it cannot predict a positive case.

### Is it better for classification or regression scenarios?

This will depend on the outcome variable. I was able to find three examples where SVM was successful as either a regression or classification model. The paper by Dinesh in 21 discusses the use of SVM to successfully compare images of breast cancer screenings for early diagnosis. Han in 2020 used SVM classification model to predict a landslide by comparing historical data from fiber optic cables buried in the hillside which collected over time. Their SVM classifier was able to predict a historic landslide, suggesting that this technology can be used to predict future landslides. Otchere in 2020 published a paper in which they systematically review methods and demonstrate the SVM regression outperforms neural network for predicting the size of an oil reservoir.

### Do you agree with the recommendations?

From all the papers cited, it becomes apparent whether SVM with outperform or not, depends on the context, the dataset, the question being asked, and the outcome variable. The purpose of data science is to find the best predictive model to serve each of these elements. In my case, SVM performed the same as a random forest because the data are artificial. In real life dataset, SVM could outperform or under perform compared to random forest (or any other model), which is why we would need to test them both.

# CONCLUSION

I fitted three SVM models with three different kernels, to try to build a robust classification model from 10000 simulated sales. My outcome of order priority was re-coded into a binary variable “high” and “not high”. The linear and radial kernel SVM classifiers performed equally as well as the random forest as in the previous assignment. This was very likely because the data was simulated but had it been real data.  
From the paper assigned as well as the three I found, it becomes apparent that whether SVM with outperform or not, depends on the context, the dataset, the question being asked, and the outcome variable. Thus, a real-world dataset could have performed differently. The importance in data science is to test multiple models to find the best predictive model.

# REFERENCES

<https://pubs.aip.org/aip/acp/article-abstract/2853/1/020140/3290220/Medical-image-prediction-for-diagnosis-of-breast>

<https://www.sciencedirect.com/science/article/pii/S0013795220317737?casa_token=7mNpV7Y-e0oAAAAA:XFzGZG_q3MTiGcK4iUDAvWhDQWr5N4dVUMv9uUnaY0X-hBFILf51nNUpjhIu_ITKlMado_DaIlc>

<https://www.sciencedirect.com/science/article/pii/S0920410520312365?casa_token=CSQrsIXon58AAAAA:46QVqJJgXi7bmqxULhcvs4YZcG2JTbFs9GdoElcflGKzGerMQf4Kf3Jp12aRfMWfqQ0ns41TW-4>