

TP BDM

Module :

Big Data Mining

Groupe :

2CS SIL1

Membres de l'équipe :

BENKHELIFA Bouchra

YOUSFI Sarah

Contents

1	INTRODUCTION	3
2	SVM	4
2.1	IRIS	4
2.2	Airline Passengers Satisfaction	6
3	CONSLUSION	8

Chapter 1

INTRODUCTION

Le Support Vector Machine (SVM) est un algorithme de machine learning utilisé pour classer des données en fonction de leurs caractéristiques. L'objectif de ce TP est de comprendre comment fonctionne un SVM en l'appliquant à un problème simple de classification. Nous allons utiliser un jeu de données pour entraîner le modèle, ajuster ses paramètres et évaluer sa performance en termes de précision.

Chapter 2

SVM

2.1 IRIS

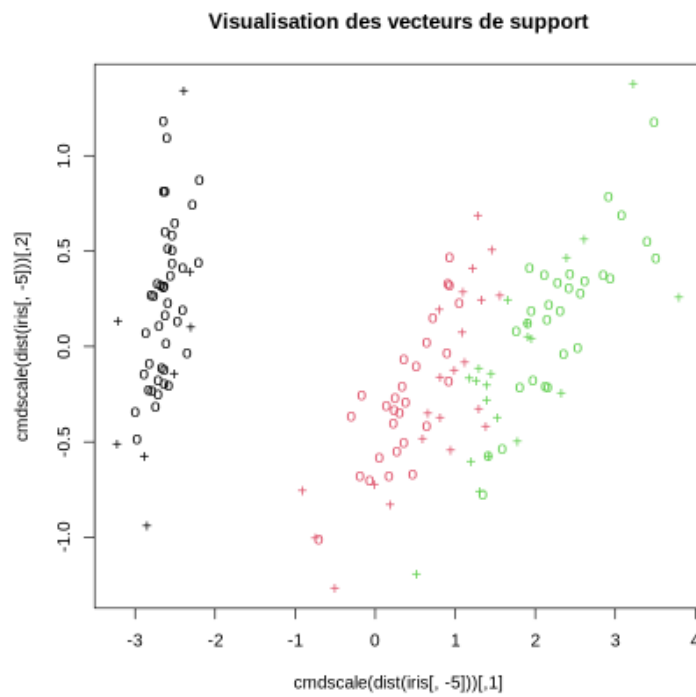


Figure 2.1: Visualization

	setosa	versicolor	virginica
setosa	14	0	0
versicolor	0	17	1
virginica	0	0	13

A data.frame: 5 × 6

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Pred
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<fct>
1	5.1	3.5	1.4	0.2	setosa	setosa
2	4.9	3.0	1.4	0.2	setosa	setosa
3	4.7	3.2	1.3	0.2	setosa	setosa
5	5.0	3.6	1.4	0.2	setosa	setosa
11	5.4	3.7	1.5	0.2	setosa	setosa

Figure 2.2: Résultat

2.2 Airline Passengers Satisfaction

Le jeu de données comporte **plus de 25 000 observations**. Toutefois, l'exécution d'un SVM sur l'ensemble de ces données s'est révélée impossible sur RStudio, en raison des limitations de nos machines locales. En alternative, nous avons tenté d'utiliser Google Colab avec un GPU T4 et 12,7 Go de RAM, mais la session a échoué (crash). Afin d'obtenir des résultats exploitables, nous avons finalement réduit le jeu de données à **10 000**

```
library(stats)
library(e1071)
test <- read.csv("test.csv")
data <- test[1:10000, ]
data <- na.omit(data)
data[sapply(data, is.character)] <- lapply(data[sapply(data, is.character)], as.factor)
set.seed(123)
n <- nrow(data)
index <- sample(n, size = 0.7 * n)
Appren <- data[index, ]
Test <- data[-index, ]
model <- svm(satisfaction ~ ., data = Appren)
print(model)
colors <- c("red", "blue")
if (ncol(data) >= 5) {
  plot(
    cmdscale(dist(data[, -5])),
    col = colors[as.integer(data$satisfaction)],
    pch = 19,
    main = "SVM Classification Results")
}
Pred <- predict(model, newdata = Test)
Test.mod <- cbind(Test, Pred)
Confusion <- table(Actual = Test.mod$satisfaction, Predicted = Test.mod$Pred)
accuracy <- sum(diag(Confusion)) / sum(Confusion)
cat("\nAccuracy:", round(accuracy * 100, 2), "%\n")
time1 <- Sys.time()
cat("Time:", format(time1, "%Y-%m-%d %H:%M:%S"), "\n")
```

Figure 2.3: Code

```

Call:
svm(formula = satisfaction ~ ., data = Appren)

Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: radial
          cost: 1

Number of Support Vectors: 1919

Warning message in dist(data[, -5]):
"NA's introduced by coercion"

Accuracy: 93.48 %
Time: 2025-04-29 11:37:16

```

Figure 2.4: SVM Model

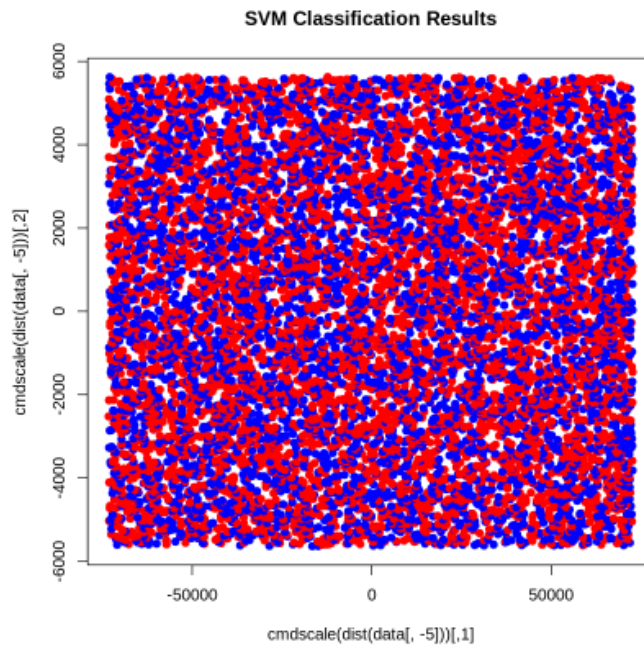


Figure 2.5: Visualization

Chapter 3

CONSLUSION

Ce TP a permis de mieux comprendre le fonctionnement du SVM et de voir comment il peut être utilisé pour résoudre un problème de classification. Nous avons appris à ajuster les paramètres du modèle et l'environnement d'exécution pour avoir des résultats exploitables .