M.M.NAVADEEP 99220041253

# Exp no : 9 SUPPORT VECTOR MACHINE

**Date :** 27/09/2023

# AIM :

To implement the support vector machine algorithm by using R programming in R studio.

# PROCEDURE :

1. Import the packages.
2. Make them in column wise.
3. Encode the target feature as a factor.
4. Spilt the dataset into training set and Test set.
5. Feature scaling.
6. Fitting SVM into training set.
7. Predicting the test results.
8. Making the confusion matrix.
9. Plotting the traing dataset as results.

# CONCEPT APPLIED :

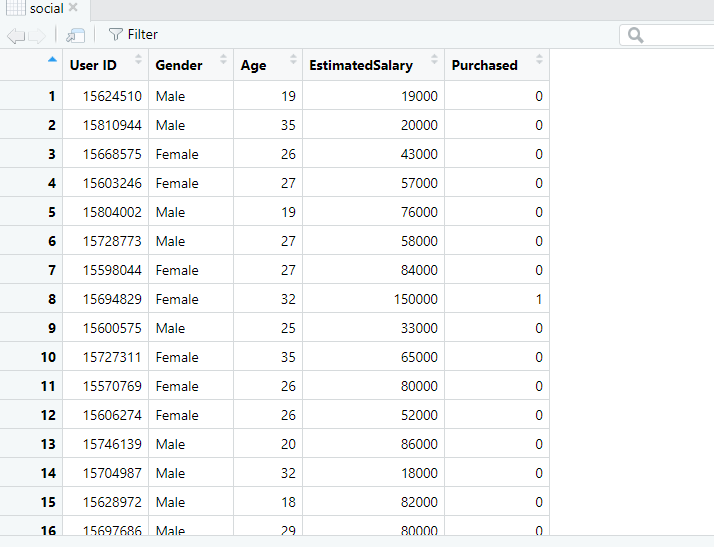
A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new examples.

# PROGRAM CODE :

library(readr)

social <- read\_csv("C:\\Users\\Asus\\Downloads\\social.csv") View(social)

# OUTPUT :



# PROGRAM CODE :

# Taking columns 3-5 dataset = dataset[3:5]

# OUTPUT :



**PROGRAM CODE :**

# Encoding the target feature as factor

dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))

# OUTPUT :



**PROGRAM CODE :**

# Splitting the dataset into the Training set and Test set install.packages('caTools')

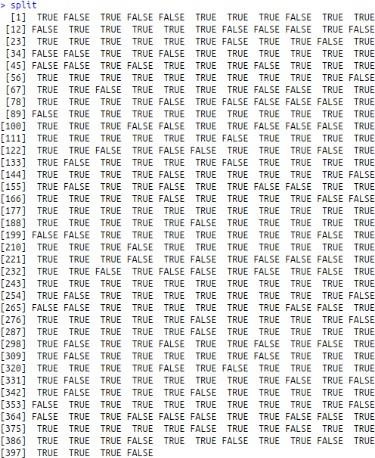
library(caTools)

set.seed(123)

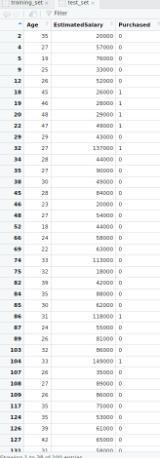
split = sample.split(dataset$Purchased, SplitRatio = 0.75)

training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)

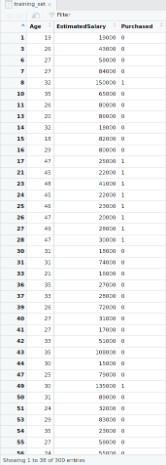
# OUTPUT :



Testing dataset :



Training dataset:



# PROGRAM CODE :

# Feature Scaling

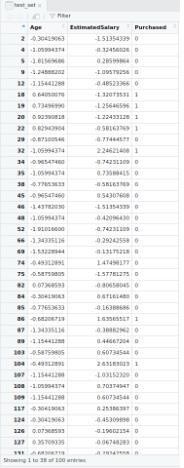
training\_set[-3] = scale(training\_set[-3]) test\_set[-3] = scale(test\_set[-3])

# OUTPUT :

Training dataset:



Testing dataset :



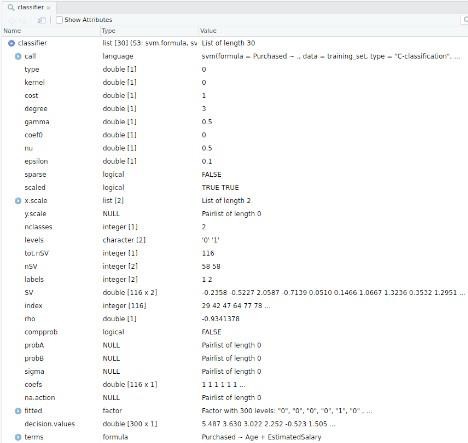
# PROGRAM CODE :

# Fitting SVM to the Training set install.packages('e1071') library(e1071)

classifier = svm(formula = Purchased ~ .,

data = training\_set, type = 'C-classification', kernel = 'linear')

# OUTPUT :

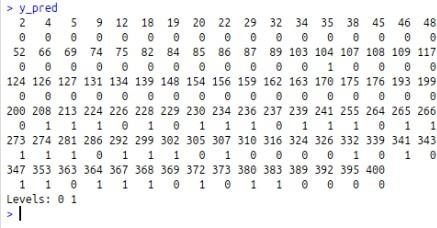


**PROGRAM CODE :**

# Predicting the Test set results

y\_pred = predict(classifier, newdata = test\_set[-3])

# OUTPUT :



**PROGRAM CODE :**

# Making the Confusion Matrix cm = table(test\_set[, 3], y\_pred)

# OUTPUT :



**PROGRAM CODE :**

# installing library ElemStatLearn library(ElemStatLearn)

# Plotting the training data set results set = training\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3],

main = 'SVM (Training set)',

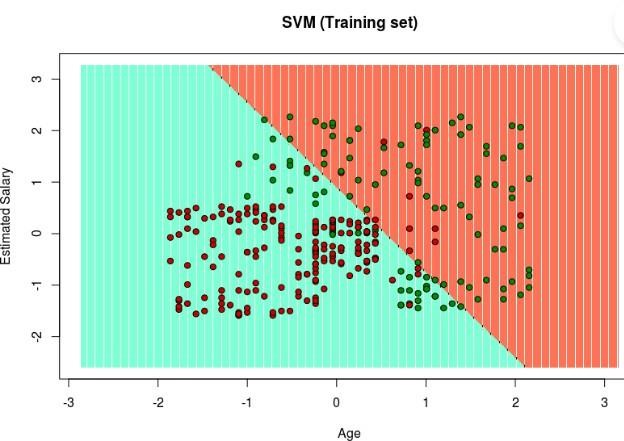
xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'coral1', 'aquamarine'))

points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

# OUTPUT :



**PROGRAM CODE :**

set = test\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3], main = 'SVM (Test set)',

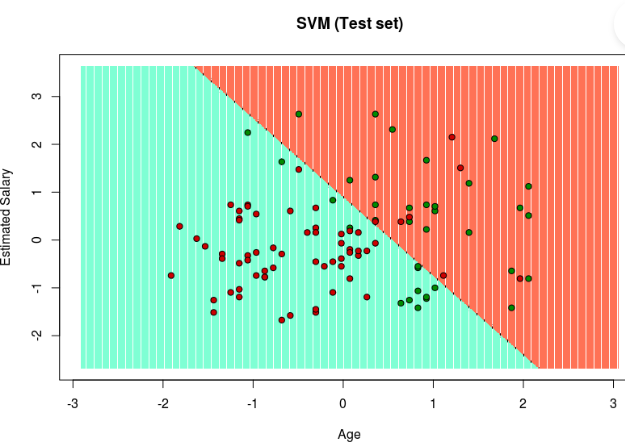
xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'coral1', 'aquamarine'))

points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

# OUTPUT :



**RESULT :**

Thus, the support vector machine algorithm was implemented by using R programming in R studio.