Tutorial 5 Memo

Dr. Niladri Chakraborty

2024-05-06

1. Write a function calculate_tip() that calculates the tip amount based on a bill amount and a tip percentage. The function should take two arguments: bill_amount (a numeric value) and tip_percent (a numeric value between 0 and 100). The function should return the tip amount as a numeric value.

```
calculate_tip <- function(bill_amount, tip_percent)
{ tip <- bill_amount * tip_percent / 100
return(tip)
}
```

2. Write a function find_missing_numbers() that takes a vector of consecutive integers (in any order) and returns a vector of the missing numbers. For example, if the input vector is c(1, 4, 3, 6, 7), the function should return c(2, 5).

```
find_missing_numbers <- function(nums) {
  complete_seq <- seq(min(nums), max(nums))
  missing_nums <- setdiff(complete_seq, nums)
  return(missing_nums)
}
```

3. Consider the 'Smarket' (S&P Stock Market Data) from the package 'ISLR'. In this dataset, raw values of the S&P 500 were obtained from Yahoo Finance and then converted to percentages and lagged. The lag values in the dataset indicate the % return for a few days previous. The variable volume is the volume of shared traded everyday.

Use the packages 'tidyverse', 'e1071', 'caret', 'ISLR', 'hrbrthemes'. Set a seed value for reproducibility of your analysis of this stock market data. Once the dataset is called, remove the very first and the last column. Use this new data for the analysis specified below.

(ii) Create a random partition of the dataset with 70:30 ratio for the training data and the test data.

(ii) Use the variable 'Today' (which is the % return for today) as the target (or dependent) variable. Use all other variables as predictors (or independent variables). Fit a SVM model.

(iii) Then use this model and the test data to predict the target variable values.

(iv) Repeat the steps (i) - (iii) for a new training and test data created by considering 50:50 random subsets of the full dataset.

(v) For these two partitions, calculate the RMSE, Rsquared and MAE values. Which partition ratio provides better SVM fit for the dataset?

(vi) Next, create another partition with 60:30 random subsets for the training and the test data, respectively. Fit another SVM model with 'Today' as the target variable and 'Lag1' and 'Lag5' as the predictors. Create a grid of Lag1 and Lag5 values between the corresponding minimum and maximum. Using the new model and the grid data, predict the % return for today. Then create a contour plot with Lag1 and Lag5 on the X and Y axis and 'Today' on the Z axis.

(vii) Then include the Lag1 and Lag5 values from the training data and the support vectors in the same contour plot.

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
```

```
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(e1071)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.3.2
```

```
library(hrbrthemes)
```

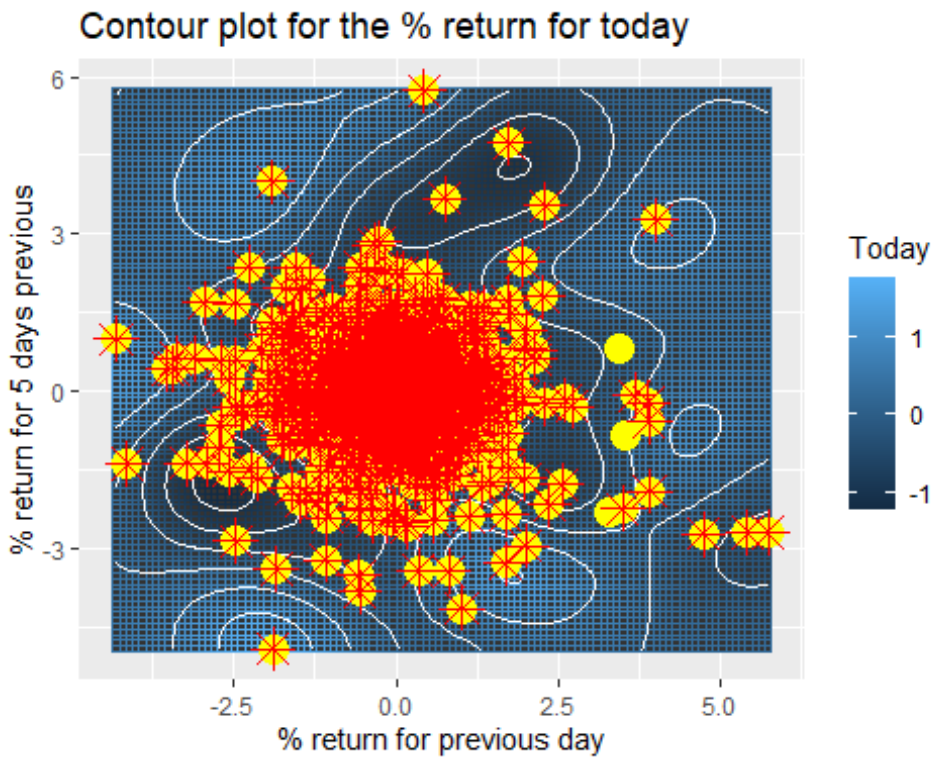
```
## Warning: package 'hrbrthemes' was built under R version 4.3.3
```

```
set.seed(123)  # for reproducibility

# Call the dataset
data <- Smarket
# Remove the first and the last column
data <- data[,-c(1,9)]

# Fitting a SVM model with training data taken as 70% random subset of the
full dataset

partition_index <- createDataPartition(data$Today, p = 0.7, list = FALSE)
train_data <- data[partition_index,]
test_data <- data[-partition_index,]

model1 <- svm(Today~.,  data = train_data)
stock_predict <- predict(model1, test_data)
result1 <- postResample(stock_predict, test_data$Today)

# Fitting a SVM model with training data taken as 50% random subset of the
full dataset
partition_index <- createDataPartition(data$Today, p = 0.5, list = FALSE)
train_data <- data[partition_index,]
```

```
test_data <- data[-partition_index,]

model2 <- svm(Today~.,  data = train_data)
stock_predict <- predict(model2, test_data)
result2 <- postResample(stock_predict, test_data$Today)

result1

##        RMSE    Rsquared         MAE
## 1.221993342 0.009501675 0.887532569

result2

##       RMSE   Rsquared        MAE
## 1.15038714 0.00489326 0.84103498
```

From the RMSE value, Rsquared value, and the MAE value, it can be said that The SVM model with 50% random subset taken as the training data has slightly better fit.

```
set.seed(123)

partition_index <- createDataPartition(data$Today, p = 0.6, list = FALSE)
train_data <- data[partition_index,]
test_data <- data[-partition_index,]

model3 <- svm(Today~ Lag1+Lag5,  data = train_data)

Lag1_seq <- seq(min(train_data$Lag1),max(train_data$Lag1), length.out = 100)
Lag2_seq <- seq(min(train_data$Lag2),max(train_data$Lag2), length.out = 100)

grid <- expand.grid(Lag1 = Lag1_seq, Lag5 = Lag2_seq)
grid$Today <- predict(model3, grid)

svr_index <- model3$index

support_vectors <- train_data[svr_index,]

ggplot(data = grid, aes(x = Lag1, y = Lag5, color = Today))+
  geom_tile()+
  geom_contour(aes(z=Today), color = "white")+
  geom_point(data = train_data, aes(x=Lag1, y=Lag5), color = "yellow", size =
5)+
  geom_point(data = support_vectors, aes(x=Lag1, y=Lag5), color="red", shape
= 8, size = 5)+
  labs(x = "% return for previous day", y = "% return for 5 days previous",
title = "Contour plot for the % return for today")
```


Contour plot for the % return for today
% return for 5 days previous
% return for previous day
6
3
0
-3
-2.5
0.0
2.5
5.0
Today
1
0
-1