Applying Support Vector Machine (SVM) Algorithm to identify S&P500 trend

Code

Hide

```
#install.packages("quantmod")
#install.packages("e1071")
library(quantmod)
library(e1071)
# Importing the dataset
startDate = as.Date("2011-01-01")
endDate = as.Date("2018-06-30")
getSymbols("^GSPC",src="yahoo",from=startDate,to=endDate)
```

```
[1] "GSPC"
```

Hide

```
dataset=data.frame(GSPC)
dim(dataset)
```

```
[1] 1886    6
```

Hide

```
str(dataset)
```

```
'data.frame':   1886 obs. of  6 variables:
 $ GSPC.Open    : num  1258 1273 1269 1276 1274 ...
 $ GSPC.High    : num  1276 1274 1278 1278 1277 ...
 $ GSPC.Low     : num  1258 1263 1265 1270 1262 ...
 $ GSPC.Close   : num  1272 1270 1277 1274 1272 ...
 $ GSPC.Volume  : num  4.29e+09 4.80e+09 4.76e+09 4.84e+09 4.96e+09 ...
 $ GSPC.Adjusted: num  1272 1270 1277 1274 1272 ...
```

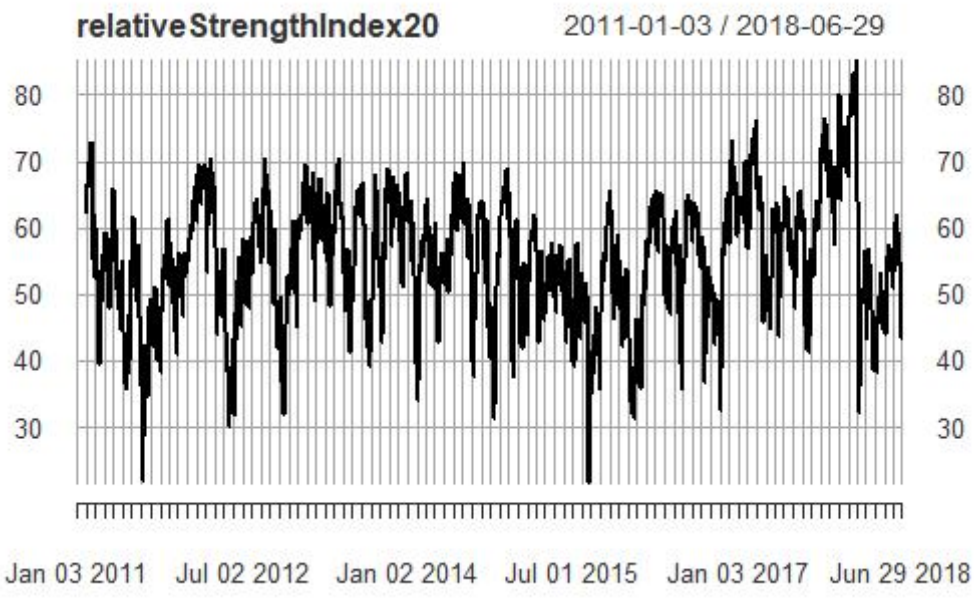
Hide

```
#RSI indicator
relativeStrengthIndex20=RSI(Op(GSPC),n=20)
summary(relativeStrengthIndex20)
```

```
     Index                rsi        
 Min.   :2011-01-03   Min.   :21.67  
 1st Qu.:2012-11-15   1st Qu.:49.28  
 Median :2014-10-01   Median :56.03  
 Mean   :2014-10-01   Mean   :55.47  
 3rd Qu.:2016-08-15   3rd Qu.:62.19  
 Max.   :2018-06-29   Max.   :85.12  
                      NA's   :20     
```

Hide

```
plot(relativeStrengthIndex20)
```



Hide

```
# Exponential Moving Average Indicator
exponentialMovingAverage20=EMA(Op(GSPC),n=20)
head(exponentialMovingAverage20)
```

```
           EMA
2011-01-03  NA
2011-01-04  NA
2011-01-05  NA
2011-01-06  NA
2011-01-07  NA
2011-01-10  NA
```

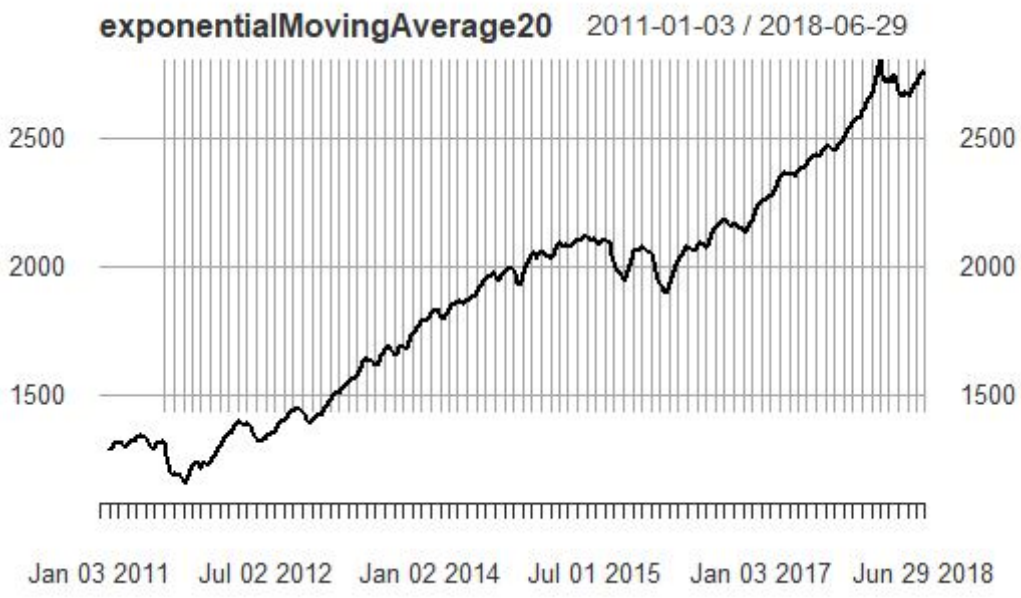
Hide

```
summary(exponentialMovingAverage20)
```

```
     Index                EMA        
 Min.   :2011-01-03   Min.   :1158  
 1st Qu.:2012-11-15   1st Qu.:1438  
 Median :2014-10-01   Median :1957  
 Mean   :2014-10-01   Mean   :1890  
 3rd Qu.:2016-08-15   3rd Qu.:2147  
 Max.   :2018-06-29   Max.   :2798  
                      NA's   :19    
```

Hide

```
plot(exponentialMovingAverage20)
```



Hide

```
# Difference in Exponential Moving Average
exponentialMovingAverageDiff <- Op(GSPC) - exponentialMovingAverage20
head(exponentialMovingAverageDiff)
```

```
           GSPC.Open
2011-01-03        NA
2011-01-04        NA
2011-01-05        NA
2011-01-06        NA
2011-01-07        NA
2011-01-10        NA
```

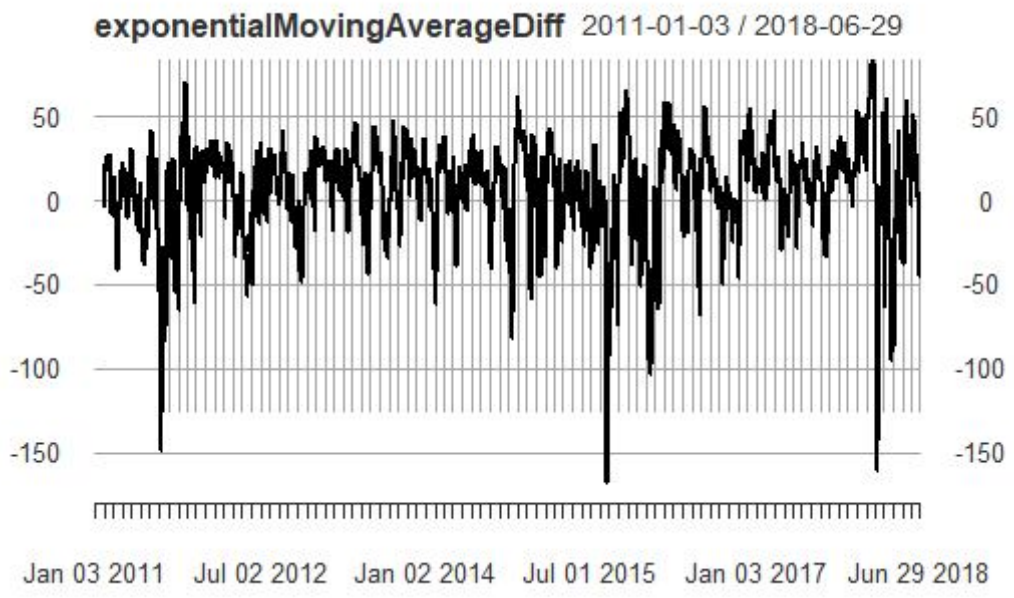
Hide

```
summary(exponentialMovingAverageDiff)
```

```
     Index              GSPC.Open       
 Min.   :2011-01-03   Min.   :-167.908  
 1st Qu.:2012-11-15   1st Qu.:  -6.288  
 Median :2014-10-01   Median :  12.592  
 Mean   :2014-10-01   Mean   :   7.432  
 3rd Qu.:2016-08-15   3rd Qu.:  25.688  
 Max.   :2018-06-29   Max.   :  83.325  
                      NA's   :19        
```

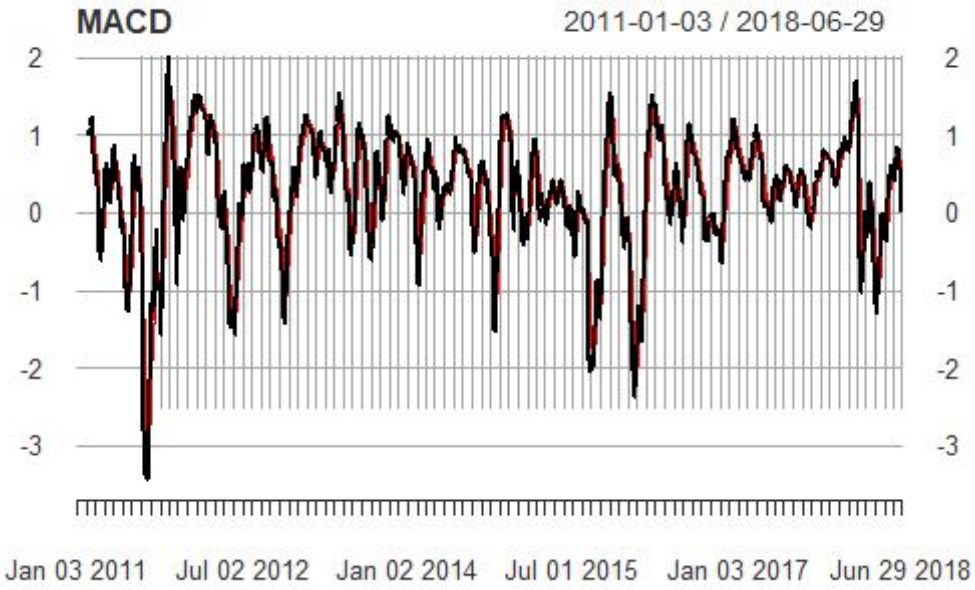
Hide

```
plot(exponentialMovingAverageDiff)
```



Hide

```
# MACD Indicator
MACD <- MACD(Op(GSPC),fast = 12, slow = 26, signal = 9, type = "EMA", histogram = TRUE)
plot(MACD)
```



Hide

```
tail(MACD)
```

```
                 macd    signal
2018-06-22 0.574743051 0.6812994
2018-06-25 0.471427824 0.6393251
2018-06-26 0.324827483 0.5764256
2018-06-27 0.224507694 0.5060420
2018-06-28 0.056802071 0.4161940
2018-06-29 0.007311846 0.3344176
```

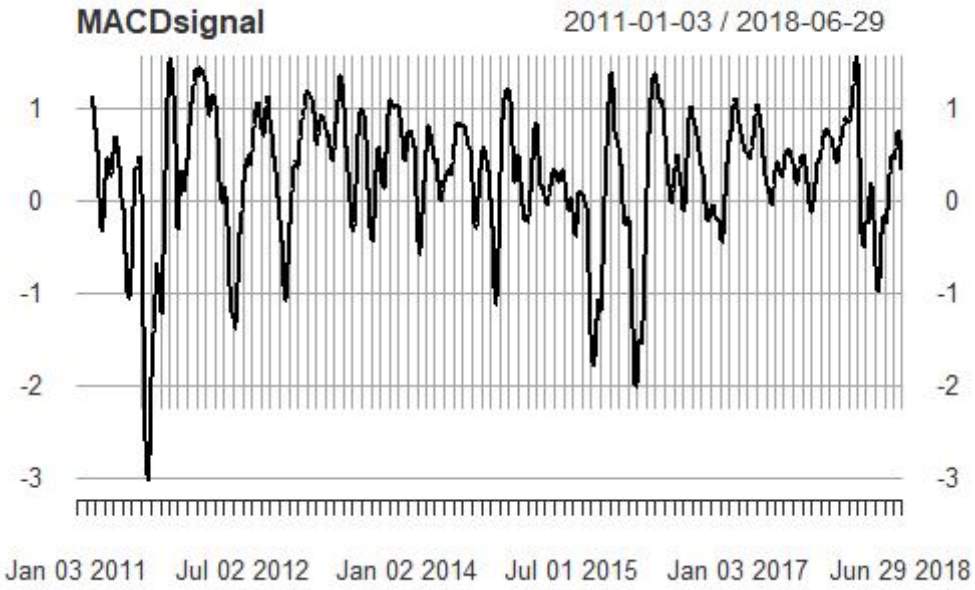
Hide

```
summary(MACD)
```

```
     Index                 macd              signal        
 Min.   :2011-01-03   Min.   :-3.44194   Min.   :-3.0188  
 1st Qu.:2012-11-15   1st Qu.:-0.07795   1st Qu.:-0.0489  
 Median :2014-10-01   Median : 0.41028   Median : 0.3901  
 Mean   :2014-10-01   Mean   : 0.28198   Mean   : 0.2801  
 3rd Qu.:2016-08-15   3rd Qu.: 0.78858   3rd Qu.: 0.7543  
 Max.   :2018-06-29   Max.   : 2.01064   Max.   : 1.5568  
                      NA's   :25         NA's   :33       
```

Hide

```
MACDsignal <- MACD[,2]
plot(MACDsignal)
```



Hide

```
# Bollinger Band indicator
BollingerBands <- BBands(Op(GSPC),n=20,sd=2)
tail(BollingerBands)
```

```
                dn     mavg       up        pctB
2018-06-22 2701.463 2754.360 2807.256  0.56078445
2018-06-25 2704.032 2755.327 2806.621  0.37926017
2018-06-26 2707.758 2756.177 2804.596  0.14831230
2018-06-27 2713.733 2757.478 2801.223  0.16821581
2018-06-28 2708.057 2756.364 2804.670 -0.09695981
2018-06-29 2709.668 2756.785 2803.902  0.18530104
```

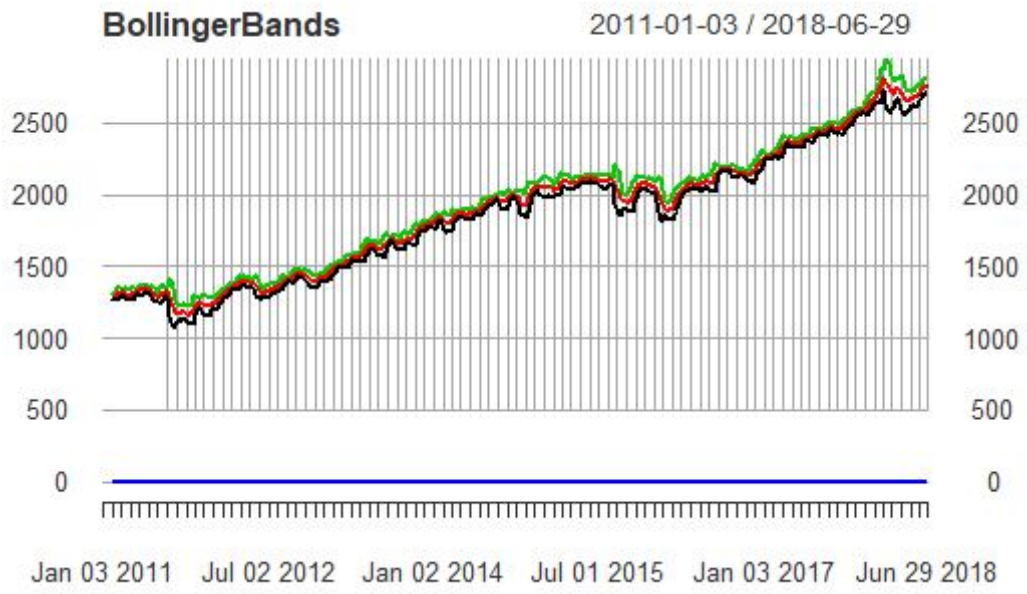
Hide

```
summary(BollingerBands)
```

```
    Index                  dn            mavg
 Min.   :2011-01-03   Min.   :1075   Min.   :1163
 1st Qu.:2012-11-15   1st Qu.:1400   1st Qu.:1438
 Median :2014-10-01   Median :1885   Median :1955
 Mean   :2014-10-01   Mean   :1844   Mean   :1890
 3rd Qu.:2016-08-15   3rd Qu.:2094   3rd Qu.:2146
 Max.   :2018-06-29   Max.   :2724   Max.   :2802
                      NA's   :19     NA's   :19
       up            pctB
 Min.   :1222   Min.   :-0.4976
 1st Qu.:1471   1st Qu.: 0.3697
 Median :2009   Median : 0.6752
 Mean   :1935   Mean   : 0.5937
 3rd Qu.:2183   3rd Qu.: 0.8456
 Max.   :2939   Max.   : 1.2709
 NA's   :19     NA's   :19
```

Hide

```
plot(BollingerBands)
```



Hide

```
# % Change BB
PercentageChngpctB <- BollingerBands[,4]
tail(PercentageChngpctB)
```

```
                pctB
2018-06-22  0.56078445
2018-06-25  0.37926017
2018-06-26  0.14831230
2018-06-27  0.16821581
2018-06-28 -0.09695981
2018-06-29  0.18530104
```

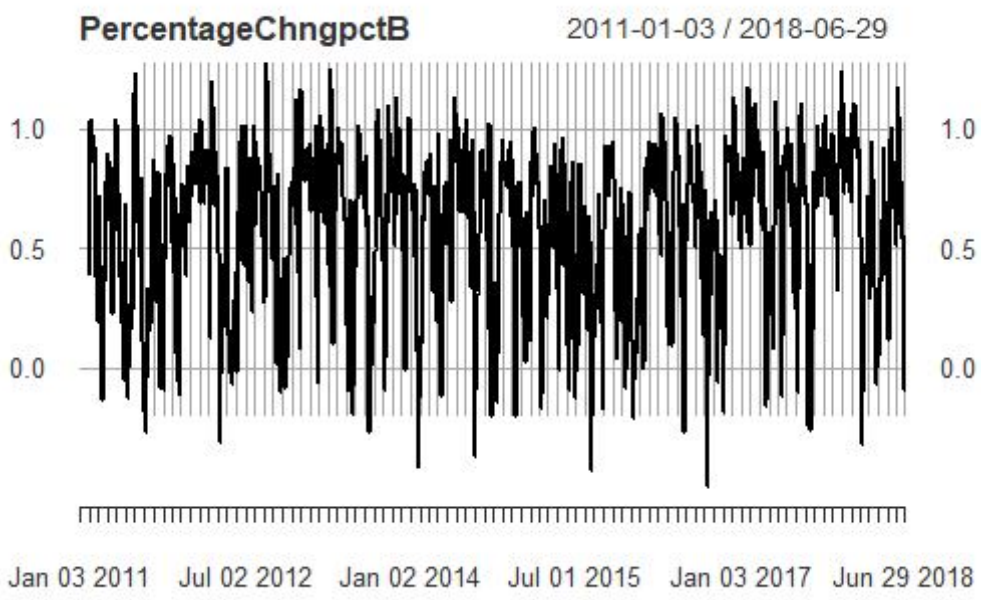
Hide

```
summary(PercentageChngpctB)
```

```
     Index                pctB          
 Min.   :2011-01-03   Min.   :-0.4976  
 1st Qu.:2012-11-15   1st Qu.: 0.3697  
 Median :2014-10-01   Median : 0.6752  
 Mean   :2014-10-01   Mean   : 0.5937  
 3rd Qu.:2016-08-15   3rd Qu.: 0.8456  
 Max.   :2018-06-29   Max.   : 1.2709  
                      NA's   :19       
```

Hide

```
plot(PercentageChngpctB)
```



Hide

```
# Price (Closes above Open = 1, Closes below Open = 0)
Price=ifelse(dataset[4]>dataset[1], 1,0)
tail(Price)
```

```
           GSPC.Close
2018-06-22          0
2018-06-25          0
2018-06-26          1
2018-06-27          0
2018-06-28          1
2018-06-29          0
```

Hide

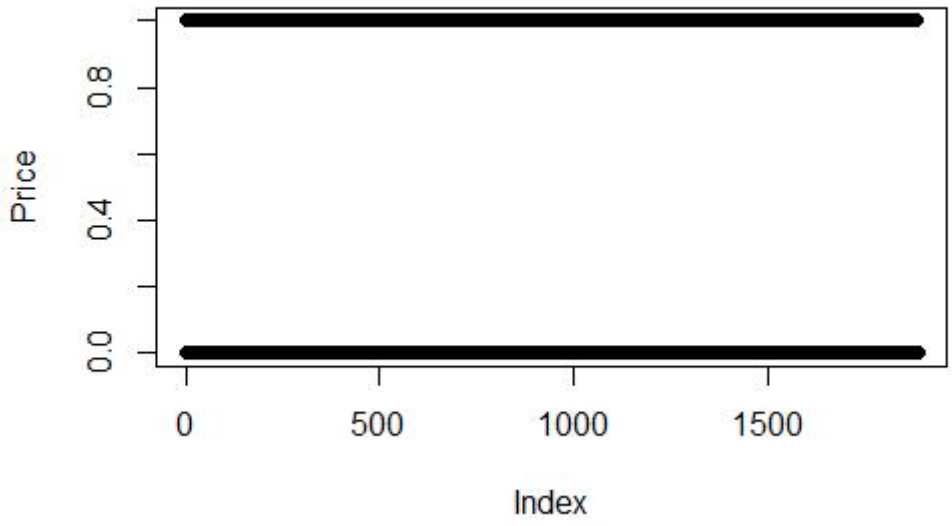
```
tail(dataset)
```

	GSPC.Op… <dbl>	GSPC.High <dbl>	GSPC.L… <dbl>	GSPC.Close <dbl>	GSPC.Volume <dbl>	GSPC.Adjusted <dbl>
2018-06-22	2760.79	2764.17	2752.68	2754.88	5450550000	2754.88
2018-06-25	2742.94	2742.94	2698.67	2717.07	3655080000	2717.07
2018-06-26	2722.12	2732.91	2715.60	2723.06	3555090000	2723.06
2018-06-27	2728.45	2746.09	2699.38	2699.63	3776090000	2699.63
2018-06-28	2698.69	2724.34	2691.99	2716.31	3428140000	2716.31
2018-06-29	2727.13	2743.26	2718.03	2718.37	3565620000	2718.37

6 rows

Hide

```
plot(Price)
```



Hide

```
dataset1<-data.frame(relativeStrengthIndex20, exponentialMovingAverage20, MACDsignal, Percentage
ChngpctB, Price)
# Size of Data
str(dataset1)
```

```
'data.frame':   1886 obs. of  5 variables:
 $ rsi       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ EMA       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ signal    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ pctB      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ GSPC.Close: num  1 0 1 0 0 0 1 1 0 1 ...
```

Hide

```
dim(dataset1)
```

```
[1] 1886    5
```

Hide

```
#Checking for missing data
d3=dataset1
for(i in 1:ncol(d3))
   {
    print(colnames(d3[i]))
    print(sum(is.na(d3[i])))
   }
```

```
[1] "rsi"
[1] 20
[1] "EMA"
[1] 19
[1] "signal"
[1] 33
[1] "pctB"
[1] 19
[1] "GSPC.Close"
[1] 0
```

Hide

```
dataset1 = na.omit(dataset1)
#Checking for missing data again
dim(dataset1)
```

```
[1] 1853    5
```

Hide

```
d3=dataset1
for(i in 1:ncol(d3))
   {
    print(colnames(d3[i]))
    print(sum(is.na(d3[i])))
   }
```

```
[1] "rsi"
[1] 0
[1] "EMA"
[1] 0
[1] "signal"
[1] 0
[1] "pctB"
[1] 0
[1] "GSPC.Close"
[1] 0
```

Hide

```
colnames(dataset1)=c ("RSI20", "EMA20", "MACDsignal", "BB", "Price")
# Exploring the data set components
str(dataset1)
```

```
'data.frame':   1853 obs. of  5 variables:
 $ RSI20     : num  72.9 71.9 58.9 55.1 55.2 ...
 $ EMA20     : num  1313 1316 1316 1315 1314 ...
 $ MACDsignal: num  1.12 1.14 1.13 1.08 1.02 ...
 $ BB        : num  0.926 0.867 0.522 0.385 0.378 ...
 $ Price     : num  1 0 0 0 1 1 0 1 1 0 ...
 - attr(*, "na.action")=Class 'omit'  Named int [1:33] 1 2 3 4 5 6 7 8 9 10 ...
  .. ..- attr(*, "names")= chr [1:33] "2011-01-03" "2011-01-04" "2011-01-05" "2011-01-06" ...
```

Hide

```
# Encoding the target feature as factor
dataset1$Price=factor(dataset1$Price, levels = c(0, 1))
```

Hide

```
# Splitting the dataset into the Training set and Test set
library(caTools)
set.seed(123)
split = sample.split(dataset1$Price, SplitRatio = 0.8)
training_set = subset(dataset1, split == TRUE)
test_set = subset(dataset1, split == FALSE)
# Feature Scaling (Normalization and dropping the predicted variable)
training_set[-5] = scale(training_set[-5])
test_set[-5] = scale(test_set[-5])
# Applying Kernel SVM Model on the Training set
library(e1071)
classifier = svm(formula = Price ~ .,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'radial')
# Predicting the Test set results
predict_val = predict(classifier, newdata = test_set[-5])
# Confusion Matrix
cm = table(test_set[, 5], predict_val)
print(cm)
```

```
   predict_val
      0   1
  0   3 169
  1   2 197
```

Hide

```
# Evaluating Model Accuracy on test data set using Confusion Matrix
Model_Accuracy=(cm[1,1] + cm[2,2])/ (cm[1,1] + cm[1,2] + cm[2,1] + cm[2,2])
print("Model Accuracy is")
```

```
[1] "Model Accuracy is"
```

Hide

```
print(Model_Accuracy)
```

```
[1] 0.5390836
```