Prediction using Supervised ML

Mridul Jajodia

8/12/2021

Reading the packages and importing data required for analysis

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

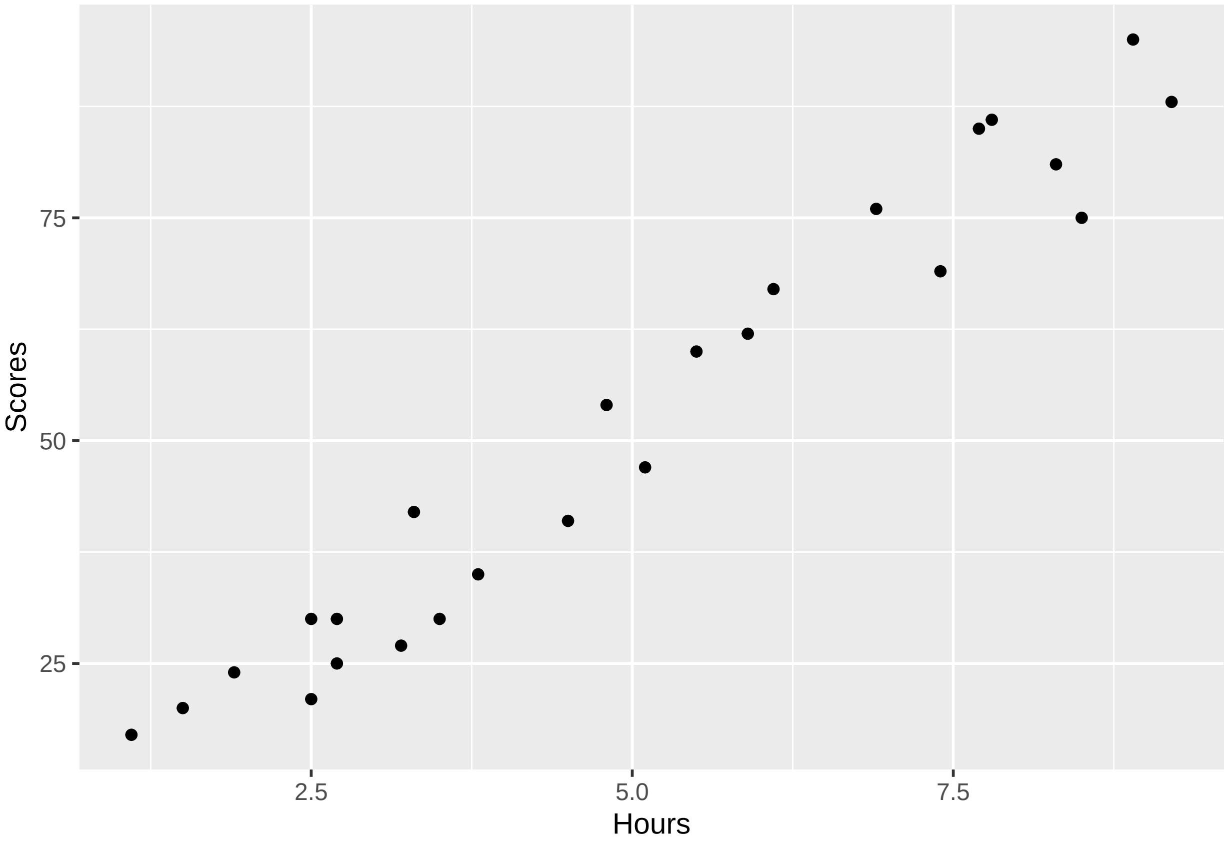
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Importing the dataset and storing it into a variable
student_data <- read.csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_score

#Showing the top 5 data from the table
head(student_data)
```

```
##   Hours Scores
## 1   2.5     21
## 2   5.1     47
## 3   3.2     27
## 4   8.5     75
## 5   3.5     30
## 6   1.5     20
```

Plotting the given data

```
#Loading ggplot2 library for plotting graph
library(ggplot2)

ggplot(data = student_data,
       aes( x = Hours, y = Scores)) +
  geom_point()
```



Finding corelation between hours and scores

```
cor(student_data$Hours,student_data$Scores)
```

```
## [1] 0.9761907
```

Splitting the dataset into training set and test set

```
install.packages("caTools")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(caTools)
split = sample.split(student_data$Hours, SplitRatio = 0.8)
trainingset = subset(student_data, split == TRUE)
testset = subset(student_data, split == FALSE)
```

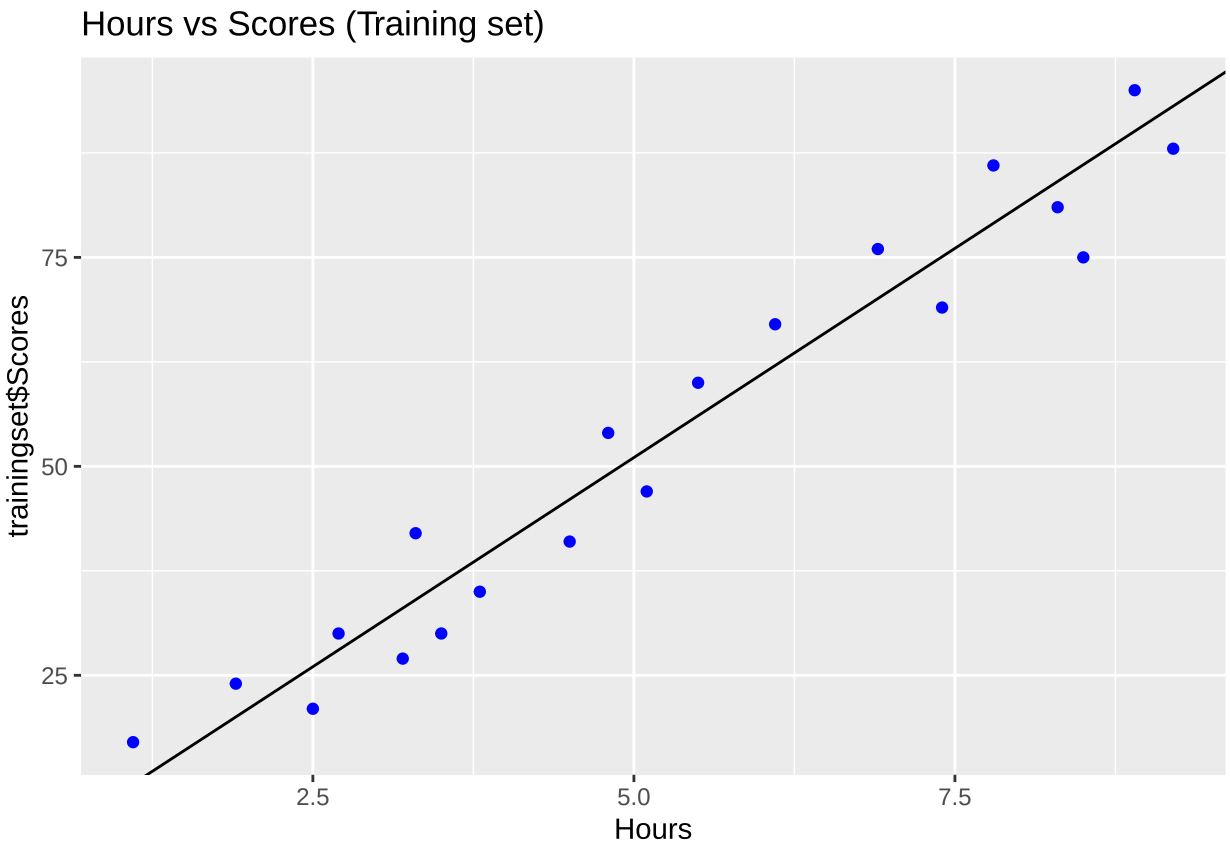
Linear regression

```
##Finding the coefficients of the data using linear model function
model = lm(formula = Scores~Hours,
           data = trainingset)
coef(model)
```

```
## (Intercept)       Hours
##    2.350530    9.695137
```

Visualising the training set results

```
ggplot(x = student_data$Hours , y = student_data$Scores) +
  geom_point(aes(x = trainingset$Hours , y = trainingset$Scores), colour = 'Blue') + geom_abline(aes(in
  ggtitle('Hours vs Scores (Training set)') +
  xlab('Hours')
```



```
  ylab('Scores')
```

```
## $y
## [1] "Scores"
##
## attr(,"class")
## [1] "labels"
```

```
#Predicting test set results
ypred = predict(model, newdata = testset)
ypred
```

```
##        6        10        11        12        17
## 16.89324 28.52740 77.00309 59.55184 26.58837
```

Predicting the score if a student studies for 9.25 hrs/day

```
new_data = data.frame(Hours = c(9.25))
predicted = predict(model, newdata = new_data)
predicted
```

```
##        1
## 92.03055
```

Conclusion - if a student studies for 9.25 hrs then the student will score nearly 92.67