**The Spark Foundation Task 2: Supervised Machine Learning**

Simple Linear Regression

In this regression task we will predict the percentage of marks that a student is expected to score based upon the number of hours they studied. This is a simple linear regression task as it involves just two variables**.**

> # View data from remote link

> Task2=read.csv(url("http://bit.ly/w-data"))

> View(Task2)

> x <- Task2$Hours

> y <- Task2$Scores

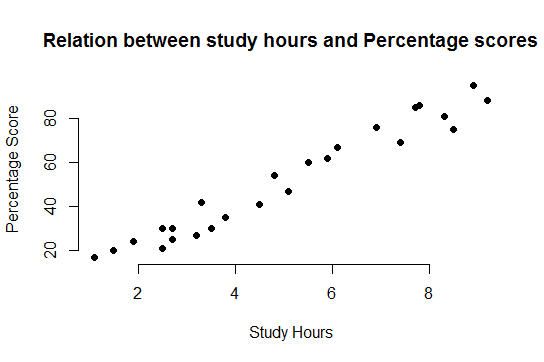
>

> #Plot scatter plot to check relation between the data

> plot(x, y, main = "Relation between study hours and Percentage scores",

+ xlab = "Study Hours", ylab = " Percentage Score",

+ pch = 19, frame = FALSE)



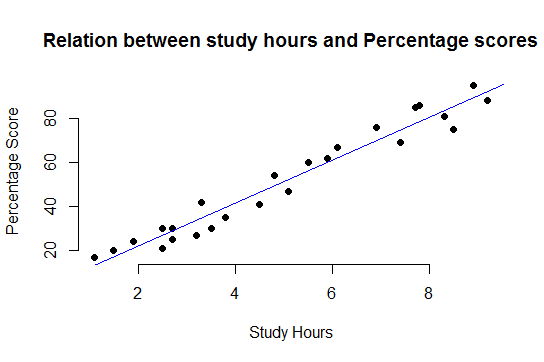
> # Add regression line

> plot(x, y, main = "Relation between study hours and Percentage scores",

+ xlab = "Study Hours", ylab = "Percentage Score",

+ pch = 19, frame = FALSE)

> abline(lm(y ~ x, data = Task2), col = "blue")



>

> #comment:From the graph above, we can clearly see that there is a positive linear relation between the number of hours studied and percentage of score.

>

> #We have split our data into training and testing sets

> n = nrow(Task2)

> n

[1] 25

> set.seed(123) # To get the same random sample each time

> train.index = sample(n,floor(0.80\*n))

>

> #Train\_data

> train.data = Task2[train.index,]

> dim(train.data)

[1] 20 2

>

> #Test\_data

> test.data = Task2[-train.index,]

> dim(test.data)

[1] 5 2

> test.data

Hours Scores

4 8.5 75

5 3.5 30

12 5.9 62

16 8.9 95

24 6.9 76

> model1=lm(Scores~Hours,data=train.data)

> summary(model1)

Call:

lm(formula = Scores ~ Hours, data = train.data)

Residuals:

Min 1Q Median 3Q Max

-7.032 -5.026 1.663 3.806 7.241

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.9179 2.5216 1.157 0.262

Hours 9.7232 0.4876 19.942 1.01e-13 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.237 on 18 degrees of freedom

Multiple R-squared: 0.9567, Adjusted R-squared: 0.9543

F-statistic: 397.7 on 1 and 18 DF, p-value: 1.013e-13

>

> #Make Prediction

> #A) Let's test our model by predicting on our testing set:

> distPred=predict(model1, test.data)

> distPred

4 5 12 16 24

85.56477 36.94899 60.28457 89.45404 70.00772

>

> #B) Let's create a dataset of actual and predicted results to check model performance:

> actuals\_preds <- data.frame(cbind(actuals=test.data$Scores, predicteds=distPred)) # make actuals\_predicteds dataframe.

> correlation\_accuracy <- cor(actuals\_preds)

> correlation\_accuracy

actuals predicteds

actuals 1.0000000 0.9529323

predicteds 0.9529323 1.0000000

> head(actuals\_preds)

actuals predicteds

4 75 85.56477

5 30 36.94899

12 62 60.28457

16 95 89.45404

24 76 70.00772

>

> #C) Let's check the performance of our model:

> sse <- sum((actuals\_preds$predicteds - actuals\_preds$actuals)^2)

> sst <- sum((mean(Task2$Scores) - actuals\_preds$actuals)^2)

> R2 <- 1-sse/sst

> R2

[1] 0.9366075

>

> #Current performance of our model is R2 = 0.9366075

>

> RMSE <- sqrt(mean((actuals\_preds$actuals - actuals\_preds$predicteds)^2))

> RMSE

[1] 6.775112

> library(MLmetrics)

> RMSE(actuals\_preds$predicteds,actuals\_preds$actuals)

[1] 6.775112

> MAPE(actuals\_preds$predicteds,actuals\_preds$actuals)

[1] 0.1074779

>

> #Conclusion - Performance of our improved model is good with R2 = 0.9366075. It means our model can explain about 93% variance in our test data.

>

> # new x=9.25 hours study

>

> # Model building

> model2=lm(y~x,data=Task2)

> summary(model2)

Call:

lm(formula = y ~ x, data = Task2)

Residuals:

Min 1Q Median 3Q Max

-10.578 -5.340 1.839 4.593 7.265

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.4837 2.5317 0.981 0.337

x 9.7758 0.4529 21.583 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.603 on 23 degrees of freedom

Multiple R-squared: 0.9529, Adjusted R-squared: 0.9509

F-statistic: 465.8 on 1 and 23 DF, p-value: < 2.2e-16

>

> # Predictions of marks for 9.25 hours study

> test2= data.frame('x'=9.25)

> distPred=predict(model2, test2)

> distPred

1

92.90985

>

> # So predicted socre is 92.9 or 93