# Linear Regression Example

## Harshad Kumar Elangovan

This is a practice sheet for a Linear Regression algorithm using a simple dataset.

Exercise: In exercise folder (same level as this notebook on github) there is hiring.csv. This file contains hiring statics for a firm such as experience of candidate, his written test score and personal interview score. Based on these 3 factors, HR will decide the salary. Given this data, you need to build a machine learning model for HR department that can help them decide salaries for future candidates. Using this predict salaries for following candidates,

2 yr experience, 9 test score, 6 interview score

12 yr experience, 10 test score, 10 interview score

```
hiring <- read.csv("hiring.csv",header = T)

head(hiring)
```

```
##   ï..experience test_score.out.of.10. interview_score.out.of.10. salary...
## 1                                   8                          9     50000
## 2                                   8                          6     45000
## 3          five                     6                          7     60000
## 4           two                    10                         10     65000
## 5         seven                     9                          6     70000
## 6         three                     7                         10     62000
```

```
#Updating the field names
colnames(hiring) <- c('experience','test_score_of_10','interview_score_of_10','salary')

head(hiring)
```

```
##   experience test_score_of_10 interview_score_of_10 salary
## 1                           8                     9  50000
## 2                           8                     6  45000
## 3       five                6                     7  60000
## 4        two               10                    10  65000
## 5      seven                9                     6  70000
## 6      three                7                    10  62000
```

Once the data is loaded, we will have to check the null values of the variables.

```
#checking for null values
hiring$experience==""
```

```
## [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

Since the experience is null. We can assume that the candidate has no experience.

```
for (i in 1:length(hiring$experience)) {
  print(hiring$experience[i])
  if(hiring$experience[i]==""){
  hiring$experience[i]='zero'
  }

}
```

```
## [1] ""
## [1] ""
## [1] "five"
## [1] "two"
## [1] "seven"
## [1] "three"
## [1] "ten"
## [1] "eleven"
```

The values of experience are in string format. We will have to update them to integer format. Since, the dataset is too small. I’m updating them manually. But if there are many rows, then a function can be created for changing to integer. One of the method is shown in the below link https://community.rstudio.com/t/convert-written-numbers-to-integers/10302/2 (https://community.rstudio.com/t/convert-written-numbers-to-integers/10302/2)

```
experience1<- c(0,0,5,2,7,3,10,11)
hiring$experienceNew <- experience1

# Checking null value of test score
is.na(hiring$test_score_of_10)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
```

```
index1 = which(is.na(hiring$test_score_of_10))
```

Since there are only few rows in dataset, we can fill an average value instead of removing the row,using median value.

```
medianval <- median(hiring$test_score_of_10,na.rm = T)
hiring$test_score_of_10[index1] <- medianval
hiring
```

```
##   experience test_score_of_10 interview_score_of_10 salary experienceNew
## 1       zero                8                     9  50000             0
## 2       zero                8                     6  45000             0
## 3       five                6                     7  60000             5
## 4        two               10                    10  65000             2
## 5      seven                9                     6  70000             7
## 6      three                7                    10  62000             3
## 7        ten                8                     7  72000            10
## 8     eleven                7                     8  80000            11
```

Now the data is set. We will carry on with the fitting the model.

```
fit<- lm(salary~experienceNew+test_score_of_10+interview_score_of_10,data = hiring)
summary(fit)
```

```
## 
## Call:
## lm(formula = salary ~ experienceNew + test_score_of_10 + interview_score_of_10, 
##     data = hiring)
## 
## Residuals:
##       1       2       3       4       5       6       7       8 
## -2350.1  -734.4  1687.0  1127.4  2729.3   851.5 -4069.1   758.4 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            17737.3     9503.9   1.866  0.13541    
## experienceNew           2813.0      281.0  10.011  0.00056 ***
## test_score_of_10        1845.7      928.6   1.988  0.11777    
## interview_score_of_10   2205.2      718.3   3.070  0.03729 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2979 on 4 degrees of freedom
## Multiple R-squared:  0.9617, Adjusted R-squared:  0.9329 
## F-statistic: 33.46 on 3 and 4 DF,  p-value: 0.002718
```

From the summary, we can see that the fit has Adjusted R-squared value to be 0.93 which is quite high. So, we can see that this fit is a good fit. From the coefficients, the experience of candidate has higher significance(99.9%) to the salary. Interview score is also significant(95%) for arriving to the salary. The test score isn’t that significant for deciding the salary of the employee.

Now, we can use this fit for predicting the salary of the candidate with parameters : 2 yr experience, 9 test score, 6 interview score.

```
#2 yr experience, 9 test score, 6 interview score.
# Predicting the intervals
predict(fit,data.frame(experienceNew=2,test_score_of_10=9,interview_score_of_10=6),interval = 
"confidence")
```

```
##        fit      lwr      upr
## 1 53205.97 46959.05 59452.89
```

```
# Predicting the salary
salary1 <- predict(fit,data.frame(experienceNew=2,test_score_of_10=9,interview_score_of_10=6
))
salary1
```

```
##        1 
## 53205.97
```

```
# 12 yr experience, 10 test score, 10 interview score
predict(fit,data.frame(experienceNew=12,test_score_of_10=10,interview_score_of_10=10),interva
l = "confidence")
```

```
##        fit      lwr      upr
## 1 92002.18 81506.56 102497.8
```

```
salary2 <- predict(fit,data.frame(experienceNew=12,test_score_of_10=10,interview_score_of_10=
10))
salary2
```

```
##        1
## 92002.18
```

So, from the predicted value, we can see that a candidate with 2 yr experience, 9 test score, 6 interview score is estimated to provide salary around 53206. For candidate with 12 yr experience, 10 test score, 10 interview score is expected to get the salary around 92002.