Simple Linear Regression

Heaven Klair

1/17/2022

Simple Linear Regression

I will use the template that I made in the pre-processing part of the project

Importing the Dataset

```
dataset = read.csv('Salary_Data.csv')
print(head(dataset))
```

```
##   YearsExperience Salary
## 1             1.1  39343
## 2             1.3  46205
## 3             1.5  37731
## 4             2.0  43525
## 5             2.2  39891
## 6             2.9  56642
```

```
# install.packages('caTools')
library(caTools)
set.seed(123)

# The dataset contains 30 observations, so let's take 20 observations in the
# training set and and 10 observations in the test set. Split Ratio = 2/3
split = sample.split(dataset$Salary, SplitRatio = 2/3)


training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

The simple linear regression package in R already takes care of the feature scaling, so we don't need feature scaling here.

Fitting Simple Linear Regression to the Training Set

```
regressor = lm(formula = Salary ~ YearsExperience,
               data = training_set)

print(summary(regressor))
```

```
##
## Call:
## lm(formula = Salary ~ YearsExperience, data = training_set)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7325.1 -3814.4   427.7  3559.7  8884.6
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        25592       2646   9.672 1.49e-08 ***
## YearsExperience     9365        421  22.245 1.52e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5391 on 18 degrees of freedom
## Multiple R-squared:  0.9649, Adjusted R-squared:  0.963
## F-statistic: 494.8 on 1 and 18 DF,  p-value: 1.524e-14
```

Interpreting the results from the summary function

Residuals section contain the summary of the residuals (the distance from the data to the fitted line). Ideally, they should be symmetrically distributed around the line; that means the Min value and Max value to be approximately the same distance from 0. Likewise, we want the first Quantile (1Q) and the third Quantile (3Q) to be equidistant from 0. Also, it is nice to have median close to 0.

The next section of Coefficients tells us about least squares estimates for the fitted line.

1. The Intercept tells us about the value of the intercept and independent variable is the slope in the equation of a straight line.
2. Std Error of the estimates and the T-value shows how p-values were calculated. These p-values test whether the estimates for the intercept and the slope are equal to 0 or not. If not, they don;t have much use in the model.
3. Pr>(|t|): These are the p-values for the estimated parameters. We know that lower the P-value is, the more significant the independent variable is going to be. This means the more impact, the more effect the independent variable is going to have on the dependent variable. Ideal number for P-value is less than 5%. So, we want the p-value for a independent variable to be less than 0.05.
4. Residual standard error: This is the square root of the denominator in the equation for F.
5. Multiple R-squared: It is just R^2. In this example, it means "Years of Experience can explain 96% of the variation in the salary."
6. Adjusted R-squared: It is the R^2 scaled by the number of parameters in the model.
7. F-statistic: It is the value for F along with Degree of Freedom(DF)
8. p-value: Here is the p-value.

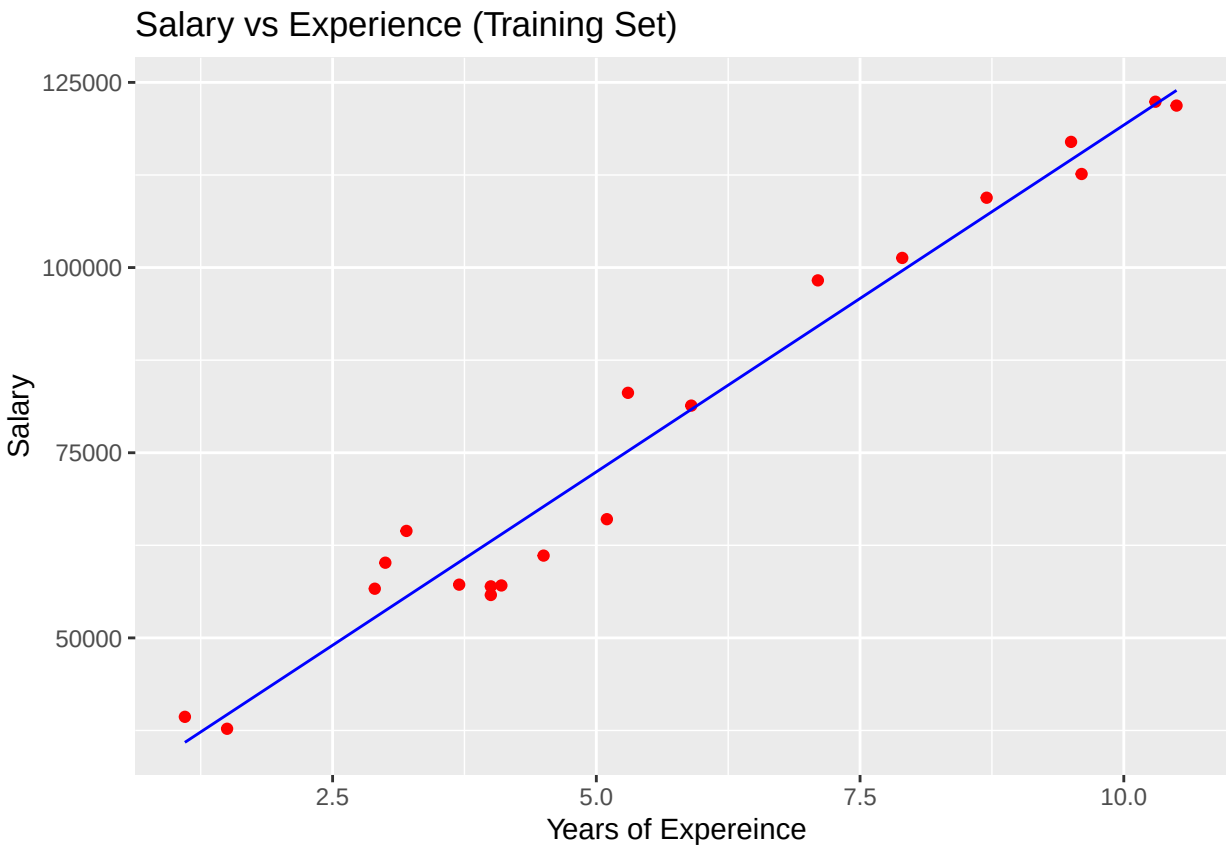
Predicting the Test Set Results

```
y_pred = predict(regressor, newdata = test_set)
print(y_pred)
```

```
##         2         4         5         8        11        16        20        21
##  37766.77  44322.33  46195.35  55560.43  62115.99  71481.07  81782.66  89274.72
##        24        26
## 102385.84 109877.90
```

Visualising the Training set results

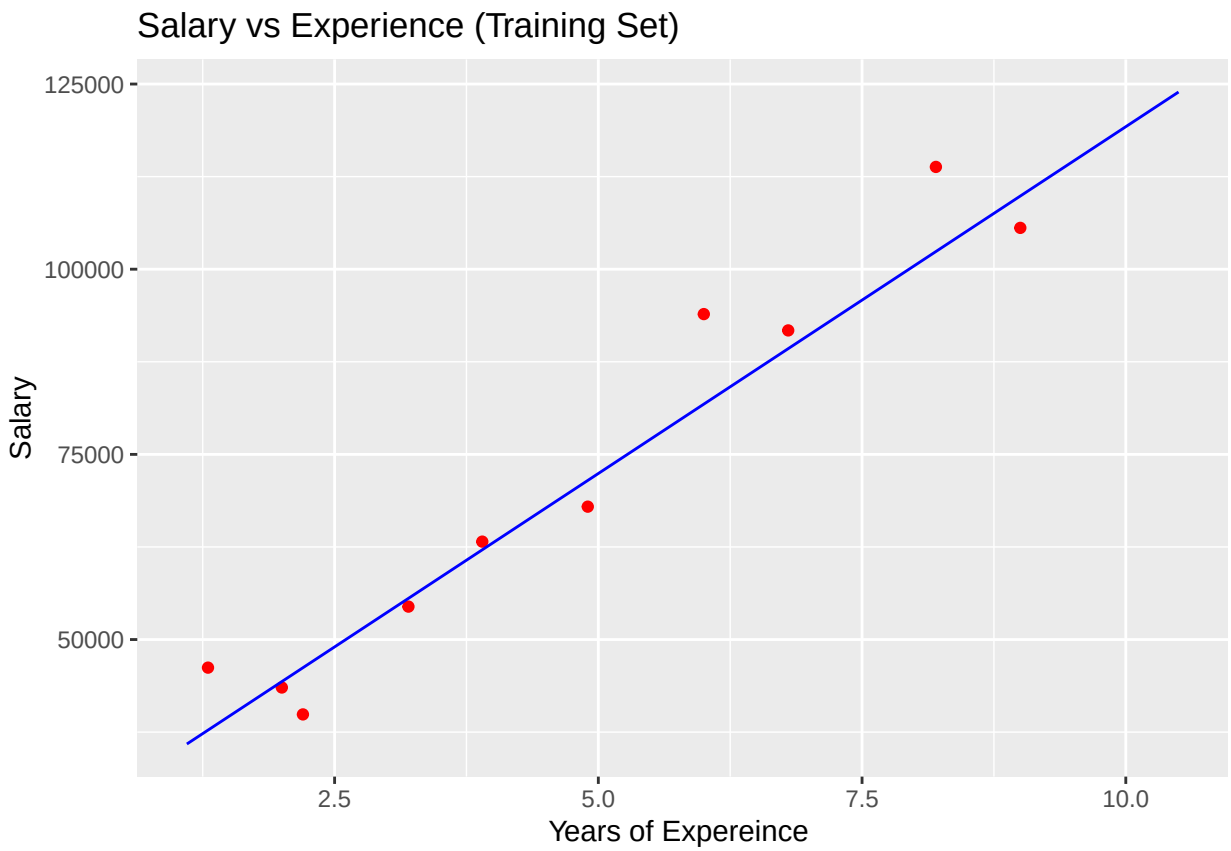
```
# install.packages('ggplot2')
library(ggplot2)

ggplot() +
  geom_point(aes(x=training_set$YearsExperience, y = training_set$Salary),
             color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata = training_set)),
            color = 'blue') +
  ggtitle('Salary vs Experience (Training Set)') +
  xlab('Years of Expereince ')+
  ylab('Salary')
```



This geom_point function will plot the scatter plot and the geom_line function will plot all the predicted salaries of the training set observations

Visualising the Test set results

```
ggplot() +
  geom_point(aes(x=test_set$YearsExperience, y = test_set$Salary),
             color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata = training_set)),
            color = 'blue') +
  ggtitle('Salary vs Experience (Training Set)') +
  xlab('Years of Expereince ')+
  ylab('Salary')
```


Salary vs Experience (Training Set)
125000
100000
75000
50000
Salary
2.5
5.0
7.5
10.0
Years of Expereince