Polynomial Linear Regression

Heaven Klair

3/30/2022

Encoding categorical data

```
dataset = read.csv('Position_Salaries.csv')
dataset = dataset[2:3]
```

We only need the second and third column from the dataset and thus, we can do that by executing the above command.

Splitting the dataset into the Training and Test set

Since the dataset is small, and we want the Training test to be as large as possible, we are not going to split the dataset into further sets.

```
# install.packages('caTools')
# library(caTools)
# set.seed(123)
# split = sample.split(dataset$DependentVariable, SplitRatio = 0.8)
# training_set = subset(dataset, split == TRUE)
# test_set = subset(dataset, split == FALSE)
```

Feature Scaling

Since, we did not create a Training or Test set, thus we do not need to do feature scaling.

Fitting Linear Regression to the Dataset

```
lin_reg = lm(formula = Salary ~.,
             data = dataset)

summary(lin_reg)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170818 -129720  -40379   65856  386545
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -195333     124790  -1.565  0.15615
## Level          80879      20112   4.021  0.00383 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 182700 on 8 degrees of freedom
## Multiple R-squared:  0.669,  Adjusted R-squared:  0.6277
## F-statistic: 16.17 on 1 and 8 DF,  p-value: 0.003833
```

Points to Notice: The level has two stars as the significance level. Thus it is a good predictor of the Salary.

Fitting Polynomial Linear Regression to the Dataset

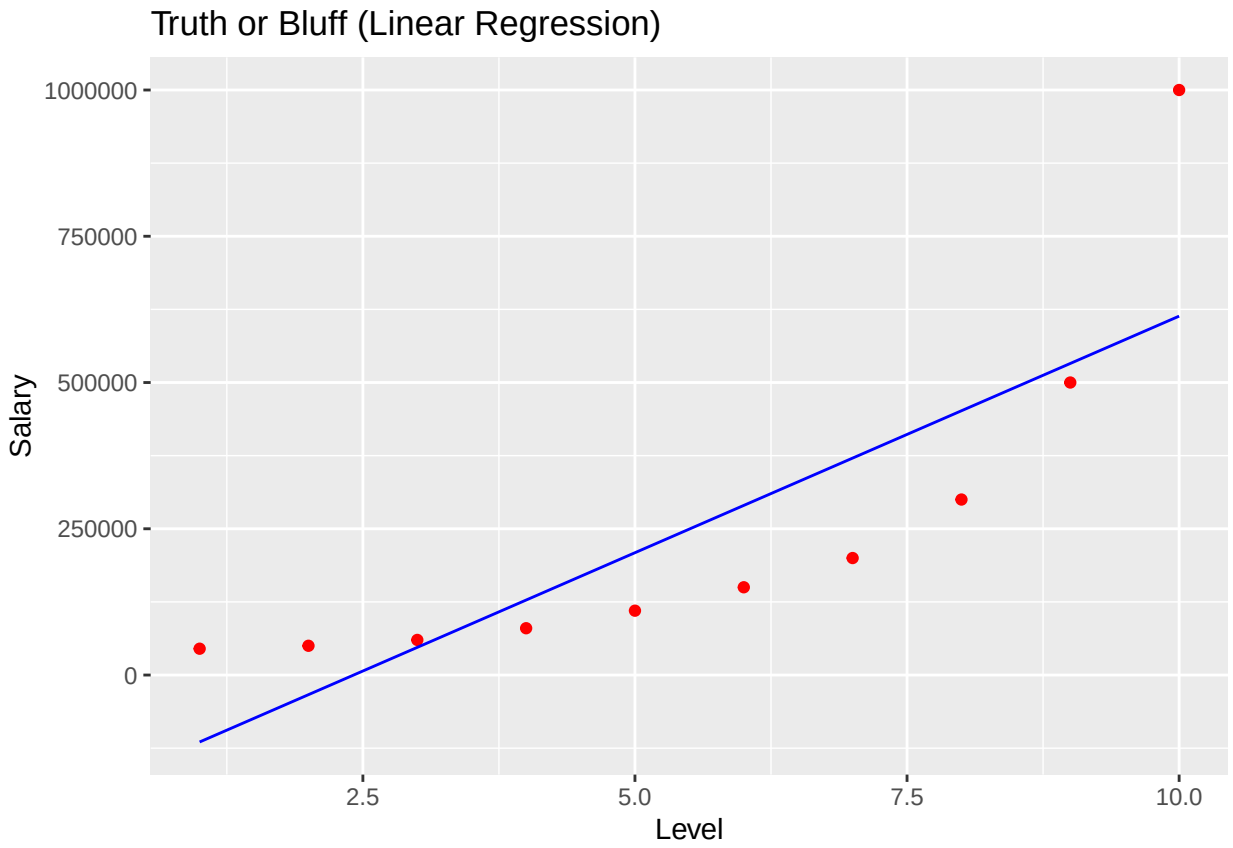
```
dataset$Level2 = dataset$Level^2
dataset$Level2 = dataset$Level^3
dataset$Level2 = dataset$Level^4
poly_reg = lm(formula = Salary ~.,
              data = dataset)

summary(poly_reg)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = dataset)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -91139 -30933   2911  36728  76318
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102368.61   53302.13   1.921   0.0963 .
## Level       -20538.64   13640.34  -1.506   0.1759
## Level2         102.67      12.17   8.433  6.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58460 on 7 degrees of freedom
## Multiple R-squared:  0.9703, Adjusted R-squared:  0.9619
## F-statistic: 114.5 on 2 and 7 DF,  p-value: 4.493e-06
```

Polynomial Regression Model is a multiple regression model that is composed of one independent variable and additional independent variables that are pulling in terms of the first independent variable.

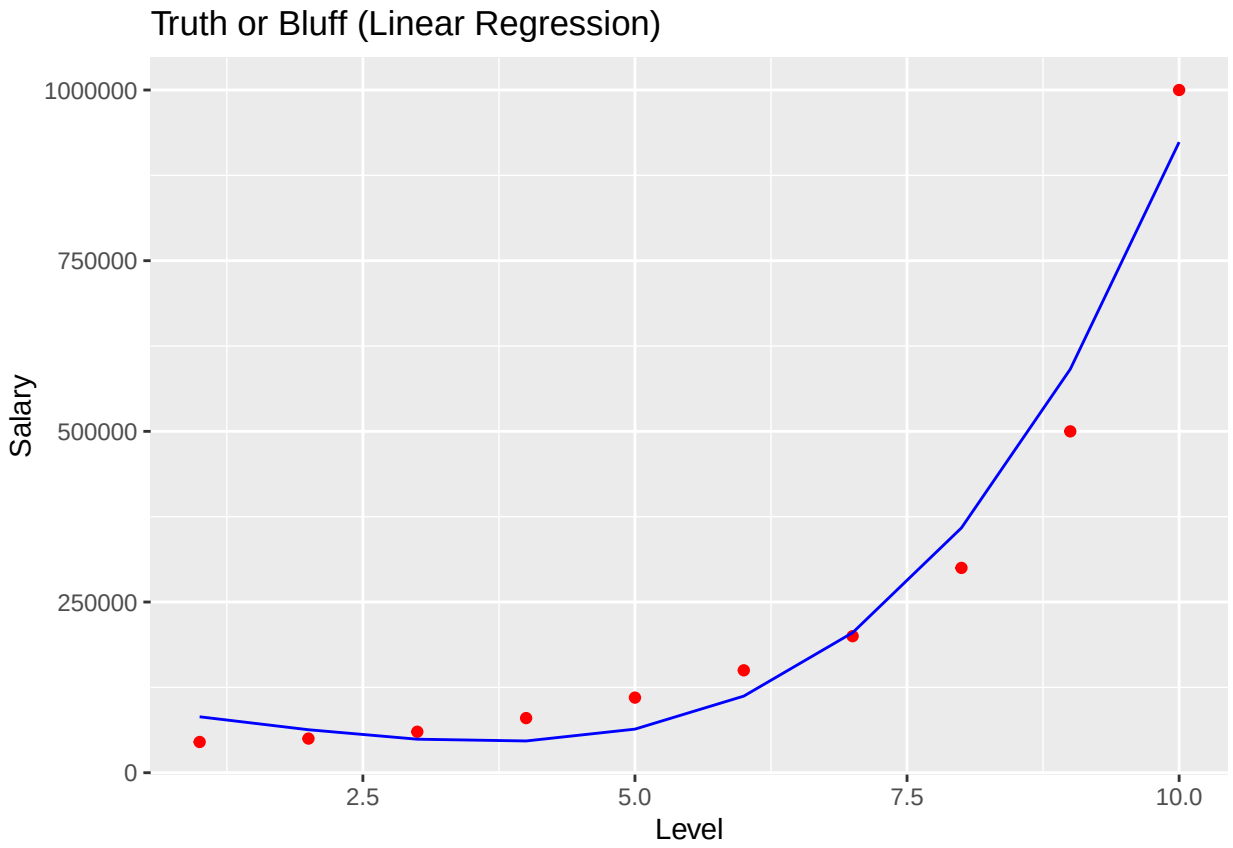
Visualising the Linear Regression Results

```
library(ggplot2)
ggplot() +
  geom_point(aes(x = dataset$Level, y = dataset$Salary),
             colour = 'red') +
  geom_line(aes(x = dataset$Level, y = predict(lin_reg, newdata = dataset )),
             colour = 'blue') +
  ggtitle('Truth or Bluff (Linear Regression)') +
  xlab('Level') +
  ylab('Salary')
```



Visualising the Polynomial Linear Regression Results

```
library(ggplot2)
ggplot() +
  geom_point(aes(x = dataset$Level, y = dataset$Salary),
             colour = 'red') +
  geom_line(aes(x = dataset$Level, y = predict(poly_reg, newdata = dataset )),
             colour = 'blue') +
  ggtitle('Truth or Bluff (Linear Regression)') +
  xlab('Level') +
  ylab('Salary')
```



Predicting a new result with Linear Regression

```
y_pred = predict(lin_reg, data.frame(Level = 6.5) )
```

Let's predict the Salary for 6.5 level. Since 6.5 is not an actual value in the dataset, we will need to create a new dataset of only one cell (one column) and this cell will contain the 6.5 level.