Support Vector Regression

Heaven Klair

4/6/2022

Regression Template

This regression Template is very useful for the next non-linear Regression models like SVR, decision tree and random forest, etc.

Importing the dataset

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

Splitting the dataset into the Training and Test set

There is no need to split the dataset into training set and test set for the Position_Salaries.csv file because the dataset is pretty small, and we want to get as much values as we can for the training purpose.

If the dataset is large enough, below is the template for splitting the dataset into training and test set.

```
# 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, we do not need to do feature scaling. However a template to feature scale the dataset is below

```
#training_set = scale(training_set)
#test_set = scale(test_set)
```

Fitting Support Vector Regression Model to the Dataset

We are going to use the SVM function because we will be taking the SVR function from the SVM package. The name of the package is "e1071". One of the important thing to notice is that we need to use the type argument in the regressor. This is because the type argument will specify if we are making the SVM model or SVR model. Since we are building a non-linear regression model, so we can choose the "eps-regression" option for type.

```
#install.packages("e1071")
library(e1071)
regressor = svm(formula = Salary ~.,
                data = dataset,
```

```
                type = 'eps-regression')

summary(regressor)
```

```
##
## Call:
## svm(formula = Salary ~ ., data = dataset, type = "eps-regression")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  6
```

Predicting a new result with Support Vector Regression Model

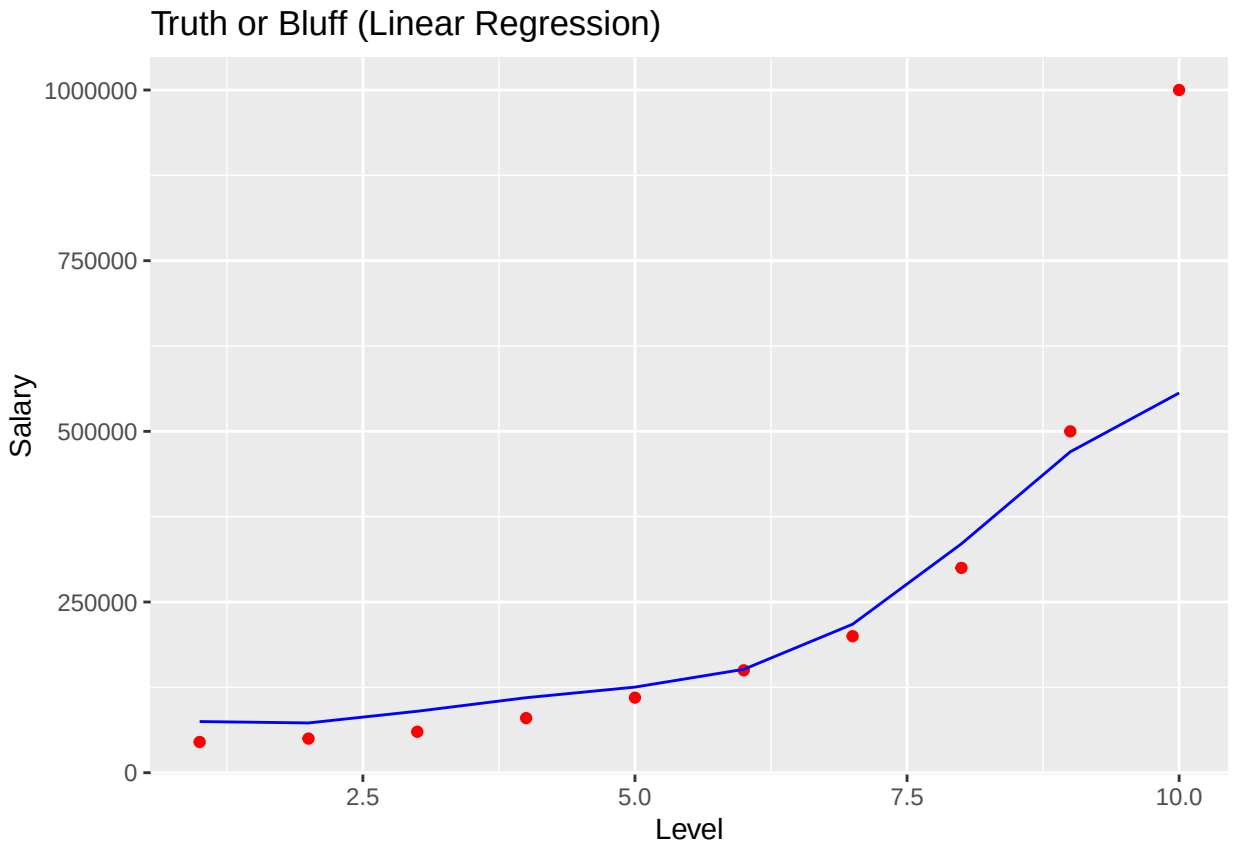
```
y_pred = predict(regressor, data.frame(Level = 6.5))
y_pred
```

```
##        1
## 177861.1
```

We can see that the predicted salary by the SVR model is $177K.

Visualising the Support Vector Regression Model Results

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


Truth or Bluff (Linear Regression)
1000000
750000
500000
250000
0
Salary
2.5
5.0
7.5
10.0
Level