Simple Linear Regression

Sunday, January 15, 2017 7:27 AM

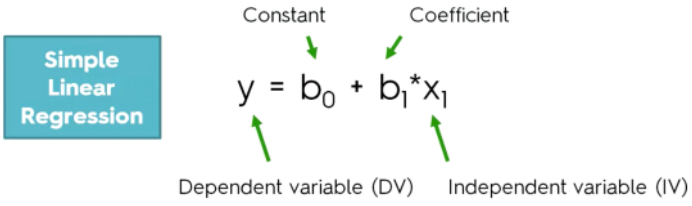
Dataset

YearsExperience	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445
3.7	57189
3.9	63218
4	55794
4	56957
4.1	57081
4.5	61111

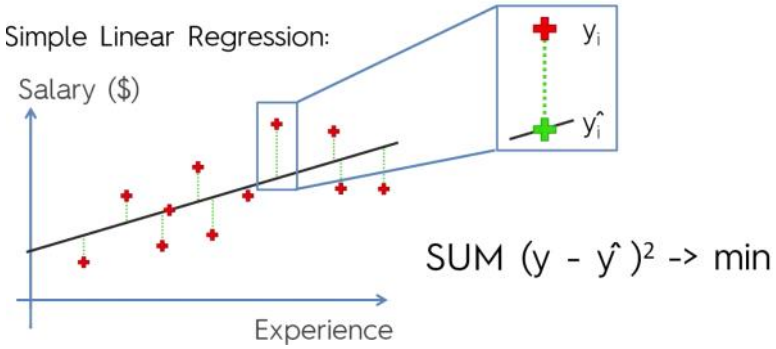
.

.

The first column contains overall experience of a person in years and the second column contains their present salary.
We need to find correlation between Salary and Experience. That is if the values are plotted in the XY plane, what is the line which approximates this pattern.



Linear Regression - Draw the best possible line that fits the data.



Here, the red cross is the actual salary of the employee. The green cross is the modelled salary for the employee. What simple regression does is it tries to minimize this difference across all points to create a line which best suits this dataset.
So we need to minimize the sum of squared errors. This is called ordinary least squares method.

Python

First step is to import and pre-process the data. This is similar to what we have done earlier. Here is a code snippet showing the steps involved.

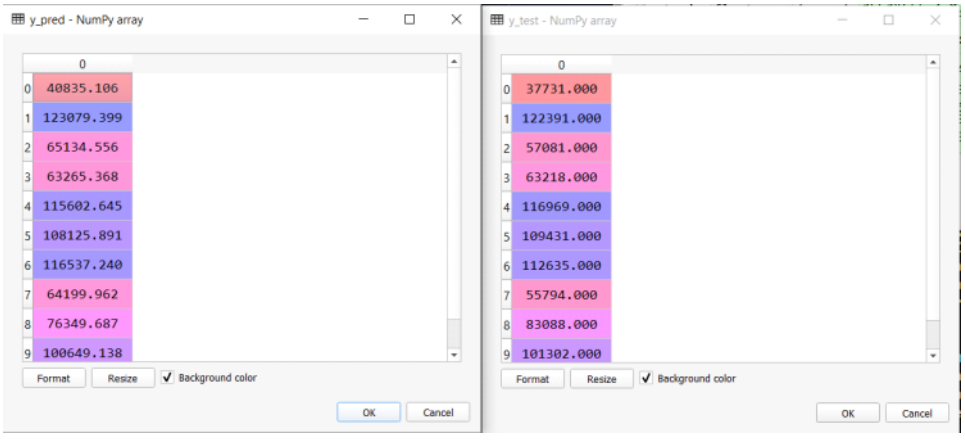
```
#Importing the libraries
Import numpy as np
Import matplotlib.pyplot as plt
Import pandas as pd

#Importing the dataset
Dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:,:-1].values
Y = dataset.iloc[:,1].values

#Splitting the dataset into the Training set and Test set
From sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)
```

Feature scaling is handled implicitly by the Simple Linear Regression model.

```
# Fitting Simple Linear Regression model to training set.
From sklearn.linear_model import LinearRegression
Regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Here, regressor is a Machine Learning model which learns on the training set composed of X_train and y_train, to understand t he correlation between Experience and Salary.

```
# Predicting the Test Set Results
y_pred = regressor.predict(X_test)
```



Plot the points and the regression line to get a visual representation of the outcome.

Here, the correlations are clear and the model has been able to make the right determination of correlations. This model can be used as a good estimate for predicting salary given experience.

R

Importing the dataset and preprocessing

```
# Importing the dataset
dataset = read.csv('Salary_Data.csv')
```

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Salary, SplitRatio = 2/3)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

We then need to fit the linear model with the training set data. Here, we use lm of R which stands for LinearModel.
We need to specify a formulae, i.e, the dependent variables and the corresponding list of independent variable. We need to fe ed data to this model to train with which in this case is the training_set.

```
# Fitting Simple Linear Regression to the Training Set
regressor = lm(formula = Salary ~ YearsExperience,
               data = training_set)
```

To find more details about the regressor model created, we can use summary function

```
summary(regressor)
```

```
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
```

Here, we get the values of the co-efficient and their corresponding statistical significance. More stars means highly statistically significant.
The P-value, i.e Pr(>|t|) -> Lower p-value the more significant the variable is.

Next we need to plot the points and the Linear Regression model in order to identify the trend and get a visual representatio n of the data.
For this, we need to install ggplot2 library. In order to do this, use

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

Ggplot2 will get installed after installing all it's dependencies now.
To use this package in our code, we use

```
library(ggplot2)
```

Now we can visualize the Training data

```
# Visualizing the Training set results
ggplot() +
  geom_point(aes(x = training_set$YearsExperience, y = training_set$Salary),
             color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, new_data = training_set)),
            color = 'blue') +
  ggtitle('Salary Vs Experience (Training Set)') +
  xlab('Years of Experience') +
  ylab('Salary in USD')
```


```
ggplot() +
  geom_point(aes(x = test_set$YearsExperience, y = test_set$Salary), color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, new_data = training_set)), color = 'blue') +
  ggtitle('Salary Vs Experience (Test Set)') +
  xlab('Years of Experience') +
```

```
  ylab('Salary in USD')
```