Published in Towards Data Science

Gurucharan M K Follow
Jun 28, 2020 · 4 min read ★ · Listen

Machine Learning Basics: Simple Linear Regression

Learn the basic Machine Learning Program of Simple Linear Regression.

One would perhaps come across the term “**Regression**” during their initial days of Data Science programming. In this story, I would like explain the program code for the very basic “***Simple Linear Regression***” with a common example.

Overview —

In statistics, ***Linear Regression*** is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). In our example, we will go through the Simple Linear Regression.

Simple Linear Regression is of the form `y = wx + b`, where ***y*** is the dependent variable, ***x*** is the independent variable, ***w*** and ***b*** are the training parameters which are to be optimized during training process to get accurate predictions.

Let us now apply Machine Learning to train a dataset to predict the ***Salary*** from ***Years of Experience***.

Step 1: Importing the Libraries

In this first step, we shall import the ***pandas*** library that will be used to store the data in a Pandas DataFrame. The ***matplotlib*** is used to plot graphs.

Step 2: Importing the dataset

In this step, we shall download the dataset from my github repositary which contains the data as “Salary_Data.csv”. The variable ***X*** will store the “***Years of Experience***” and the variable ***Y*** will store the “***Salary***”. The `dataset.head(5)` is used to visualize the first 5 rows of the data.

```
dataset = pd.read_csv('https://raw.githubusercontent.com/mk-
gurucharan/Regression/master/Salary_Data.csv')

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

dataset.head(5)

>>
YearsExperience  Salary
1.1              39343.0
1.3              46205.0
1.5              37731.0
2.0              43525.0
2.2              39891.0
```

Step 3: Splitting the dataset into the Training set and Test set

In this step, we have to split the dataset into the Training set, on which the Linear Regression model will be trained and the Test set, on which the trained model will be applied to visualize the results. In this the `test_size=0.2` denotes that ***20%*** of the data will be kept as the ***Test set*** and the remaining ***80%*** will be used for training as the ***Training set***.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
= 0.2)
```

Step 4: Training the Simple Linear Regression model on the Training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Step 5: Predicting the Test set results

In this step, the `regressor.predict()` function is used to predict the values for the Test set and the values are stored to the variable `y_pred`.

```
y_pred = regressor.predict(X_test)
```

Step 6: Comparing the Test Set with Predicted Values

In this step, a Pandas DataFrame is created to compare the Salary values of both the original Test set (***y_test***) and the predicted results (***y_pred***).

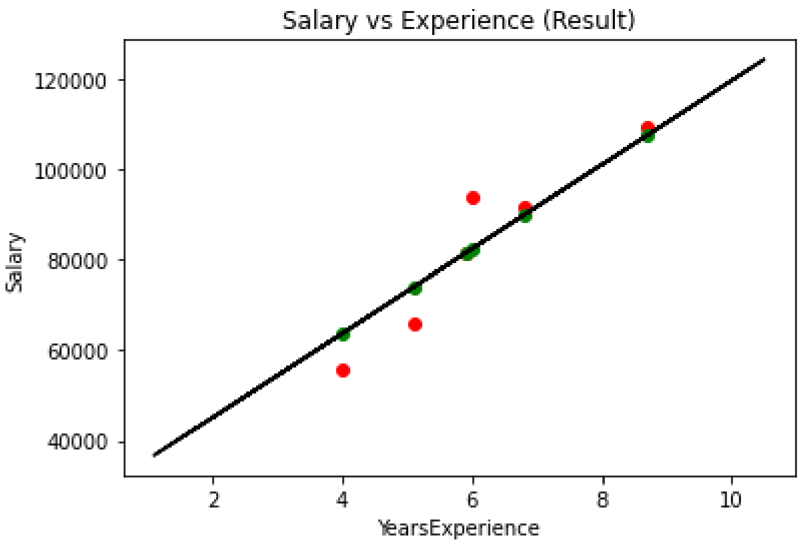
```
df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df

>>
Real Values     Predicted Values
109431.0        107621.917107
81363.0         81508.217112
93940.0         82440.849255
55794.0         63788.206401
66029.0         74047.159970
91738.0         89901.906396
```

We can see that the predicted salaries are very close to the real salary values and it can be concluded that the model has been well trained.

Step 7: Visualising the Results

In this last step, we visualize the results of the ***Real*** and the ***Predicted*** Salary values along with the Linear Regression Line on a graph that is plotted.

```
plt.ylabel('Salary')
plt.show()
```



Salary vs Experience

In this graph, the Real values are plotted in “***Red***” color and the Predicted values are plotted in “***Green***” color. The Linear Regression line that is generated is drawn in “***Black***” color.

Conclusion —

Thus in this story, we have successfully been able to build a ***Simple Linear Regression*** model that predicts the ‘Salary’ of an employee based on their ‘Years of Experience’ and visualize the results.

I am also attaching the link to my github repository where you can download this Google Colab notebook and the data files for your reference.

mk-gurucharan/Regression

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build…

github.com

- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

We will come across the more complex models of Regression, Classification and Clustering in the upcoming articles. Till then, Happy Machine Learning!