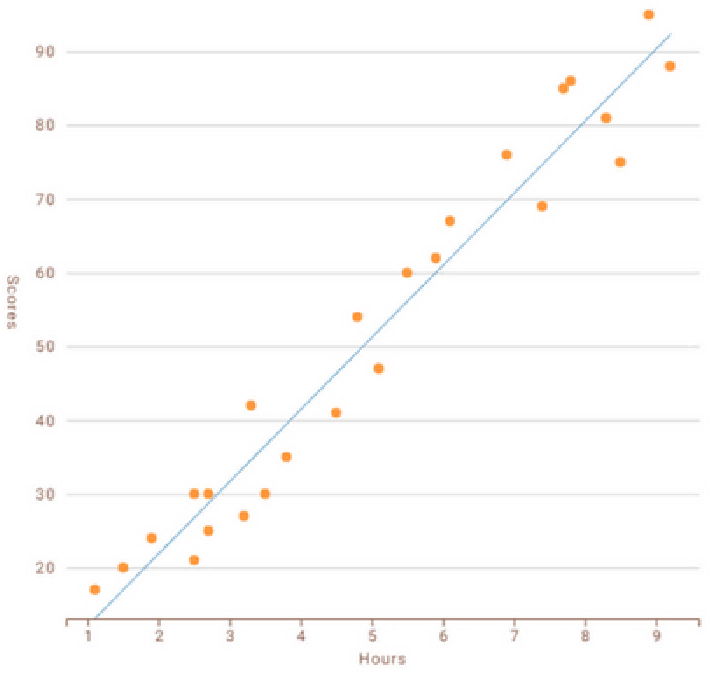
🔗Simple Linear Regression

#100DaysOfMLCode
Day 2
©Avik Jain

SIMPLE LINEAR REGRESSION



Predicting a response using a single feature.

It is a method to predict dependent variable (Y) based on values of independent variables (X). It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

Dependent variable

$$y = b_0 + b_1 x_1$$

Independent variable

How to find the best fit line?

In this regression model, we are trying to minimize the errors in prediction by finding the "line of best fit" — the regression line from the errors would be minimal. We are trying to minimize the length between the observed value (Yi) and the predicted value from our model (Yp).

Predicted value

$$y_i \quad y_p$$

Observed value

$$\min\{\text{SUM}(y_i - y_p)^2\}$$

In this regression task, we will predict the percentage of marks that a student is expected to score based upon the number of hours they studied.

Slope

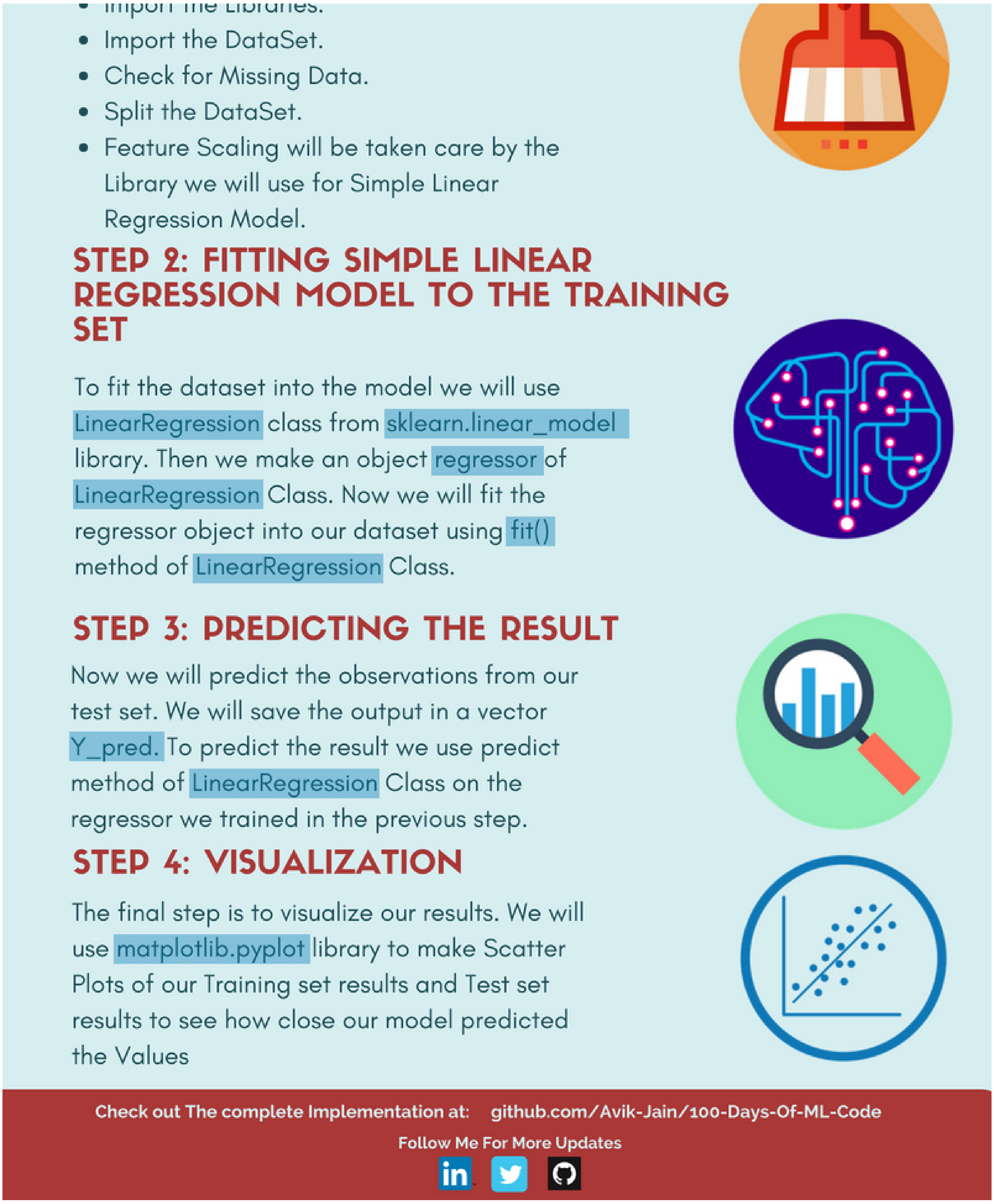
$$Score = b_0 + b_1 * hours$$

y - intercept

STEP 1: PREPROCESS THE DATA

We will follow the same steps as in my previous infographic of Data Preprocessing.

- Import the Libraries



- Import the Libraries.
- Import the DataSet.
- Check for Missing Data.
- Split the DataSet.
- Feature Scaling will be taken care by the Library we will use for Simple Linear Regression Model.

STEP 2: FITTING SIMPLE LINEAR REGRESSION MODEL TO THE TRAINING SET

To fit the dataset into the model we will use LinearRegression class from sklearn.linear_model library. Then we make an object regressor of LinearRegression Class. Now we will fit the regressor object into our dataset using fit() method of LinearRegression Class.

STEP 3: PREDICTING THE RESULT

Now we will predict the observations from our test set. We will save the output in a vector Y_pred. To predict the result we use predict method of LinearRegression Class on the regressor we trained in the previous step.

STEP 4: VISUALIZATION

The final step is to visualize our results. We will use matplotlib.pyplot library to make Scatter Plots of our Training set results and Test set results to see how close our model predicted the Values

Check out The complete Implementation at: github.com/Avik-Jain/100-Days-Of-ML-Code

Follow Me For More Updates

Step 1: Data Preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
dataset = pd.read_csv('studentscores.csv')
X = dataset.iloc[ : ,   : 1 ].values
Y = dataset.iloc[ : , 1 ].values

from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size = 1/4,
random_state = 0)
```

Step 2: Fitting Simple Linear Regression Model to the training set

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

Step 3: Predecting the Result

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

Step 4: Visualization

Visualising the Training results

```
plt.scatter(X_train , Y_train, color = 'red')
plt.plot(X_train , regressor.predict(X_train), color ='blue')
```

Visualizing the test results

```
plt.scatter(X_test , Y_test, color = 'red')
plt.plot(X_test , regressor.predict(X_test), color ='blue')
```