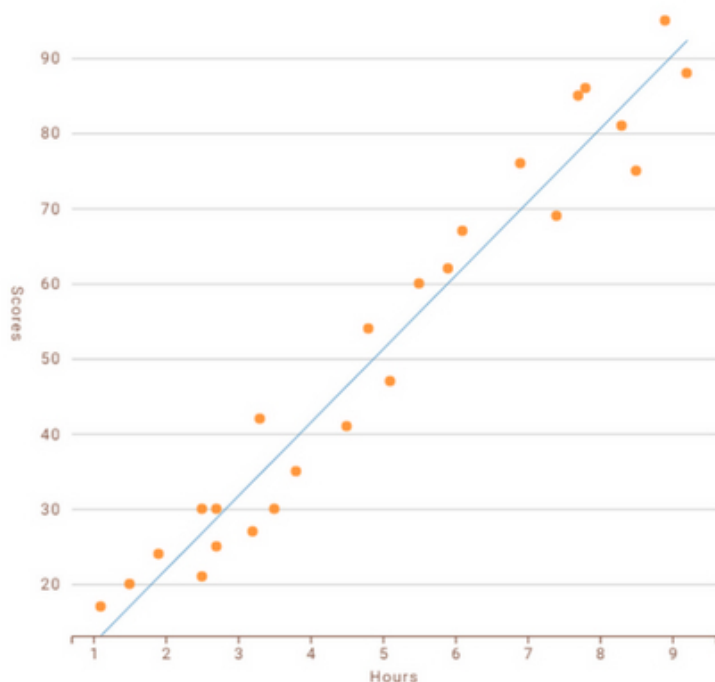


SIMPLE LINEAR REGRESSION

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).



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 (Y_i) and the predicted value from our model (Y_p).

y_i y_p
Observed Value Predicted Value

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

Dependent Variable

$$y = b_0 + b_1 x_1$$

Independent Variable

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

$$\text{Score} = b_0 + b_1 * \text{hours}$$

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 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

