Predict the marks obtained by a student based on hours of study

1. **Finding the Problem - Application**
   - To Predict
   - Predicting what mark the student will score
2. **Collecting Dataset**
   - Input: Hours of learning Output: Marks
   - Based on the number of hours students studied and how much they have scored.
3. **Load & Summarize Dataset**
   - Load Dataset from the directory & Summarize the details such as no. of rows and Columns & Content
   - Pandas - Load CSV Format Dataset
     - `dataset = pandas.read_csv('dataset.csv')`
   - No. of Rows & Columns
     - `dataset.shape`
   - Display 1st 5 rows of dataset
     - `dataset.head(5)`
4. **Segregating Dataset into X & Y**
   - iloc - It helps us select a value that belongs to a particular row or column
   - SYNTAX: `dataset.iloc[:,start_col:end_col]`
   - `X = dataset.iloc[:, :-1].values`
   - `Y = dataset.iloc[:, -1].values`
5. **Splitting Dataset to Train & Test**
   - Useful for validation
   - `train_test_split(X, Y, test_size = 0.25, random_state = 0)`
6. **Feature Scaling**

Name	Weight	Price
Orange	15	1
Apple	18	3
Banana	12	2
Grape	10	5

   - PROBLEM
     - Since both the features have different scales, there is a chance that higher weightage is given to features with higher magnitude. This will impact the performance of the machine learning algorithm and obviously, we do not want our algorithm to be biassed towards one feature.
   - SOLUTION
     - we scale our data to make all the features contribute equally to the result
   - Types

-1.184341	1.520013
-1.184341	-1.100699
0.416120	-1.100699
1.216350	0.209657
0.736212	0.471728

     - Standardization
       - $X' = \frac{X - \mu}{\sigma}$
       - Here the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.
     - Normalization
       - $X' = \frac{X - X_{min}}{X_{max} - X_{min}}$
       - Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling
7. **Algorithm**
   - Linear Regression
   - Uses 1 or More Independent Variable (X) to determine dependent variable (Y)
8. **Training**
   - Training our Model for Pre-processed Dataset
   - `model.fit(X_train, y_train)`
9. **Validation**
   - Obtaining the accuracy of the Model
   - Confusion Matrix

Actual Value	Classifier Prediction: Positive	Classifier Prediction: Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

10. **Prediction**
    - Observing how our model is classifying our new data
    - `result = model.predict(X_test)`