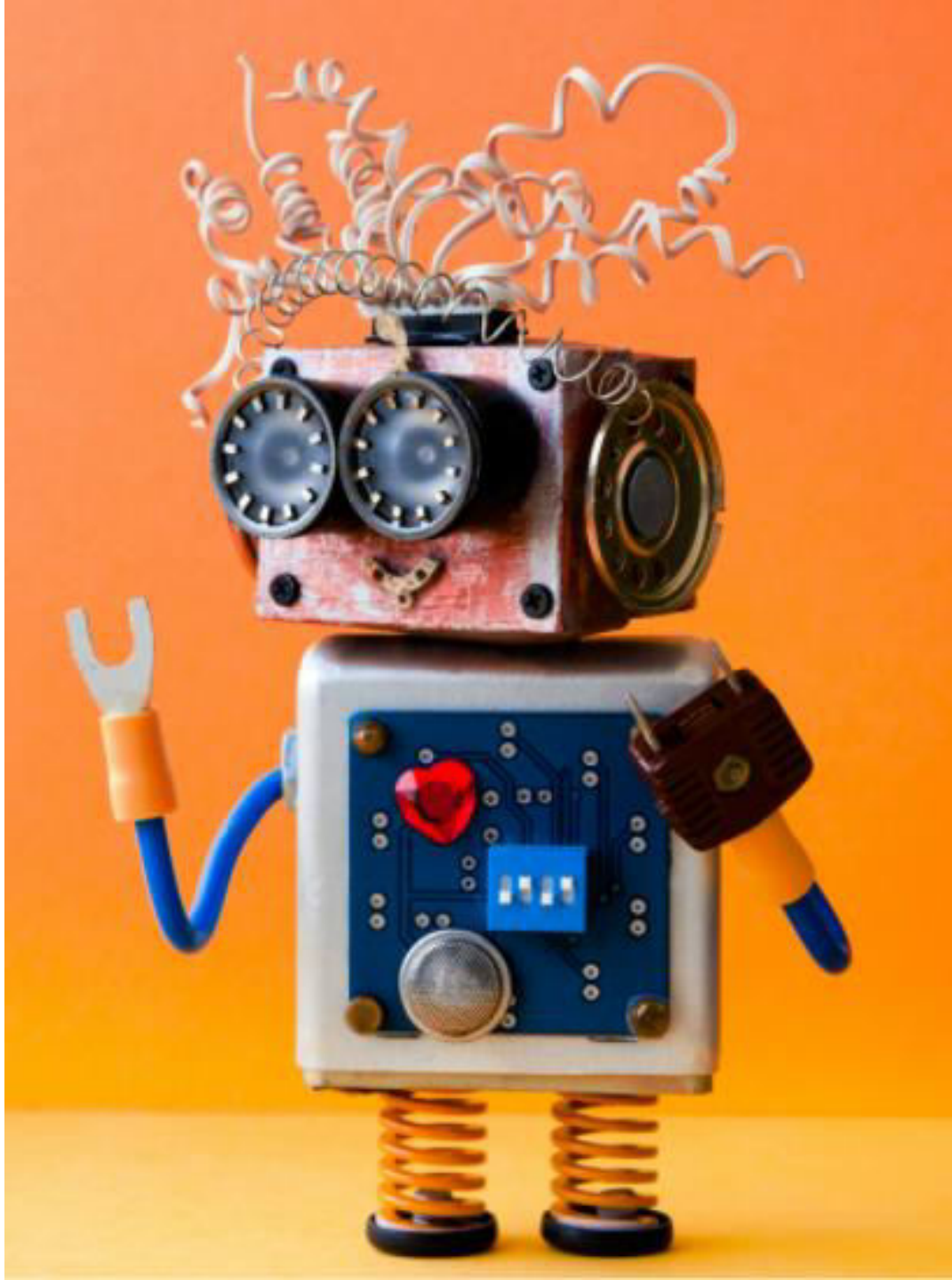
PROJECT OVERVIEW

EASY ADVANCED

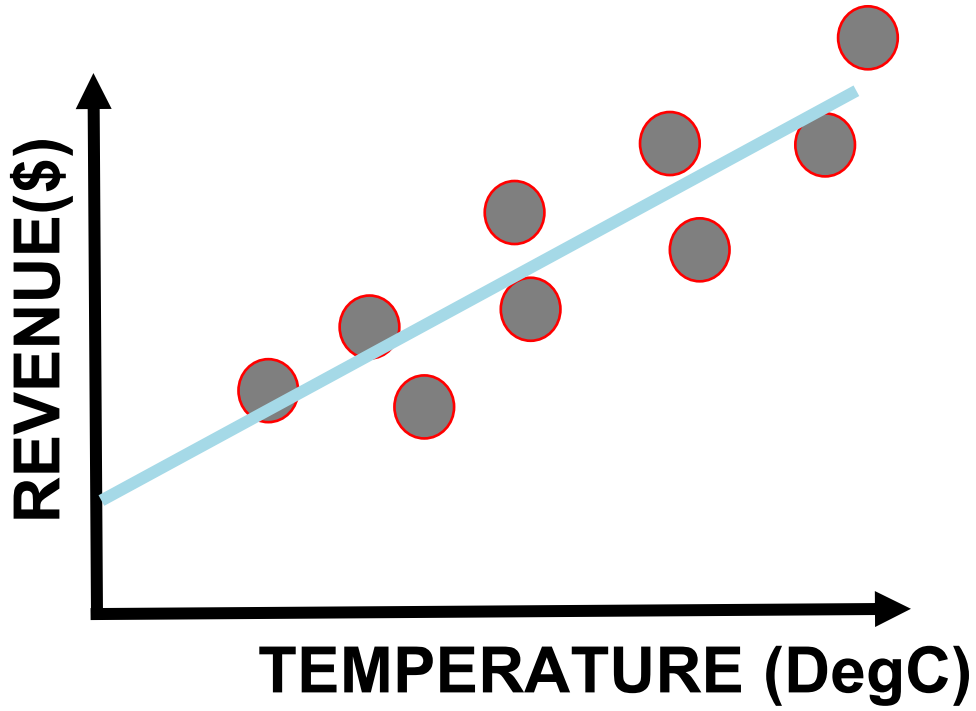


PROJECT OVERVIEW

- You own an ice cream business and you would like to create a model that could predict the daily revenue in dollars based on the outside air temperature (degC).

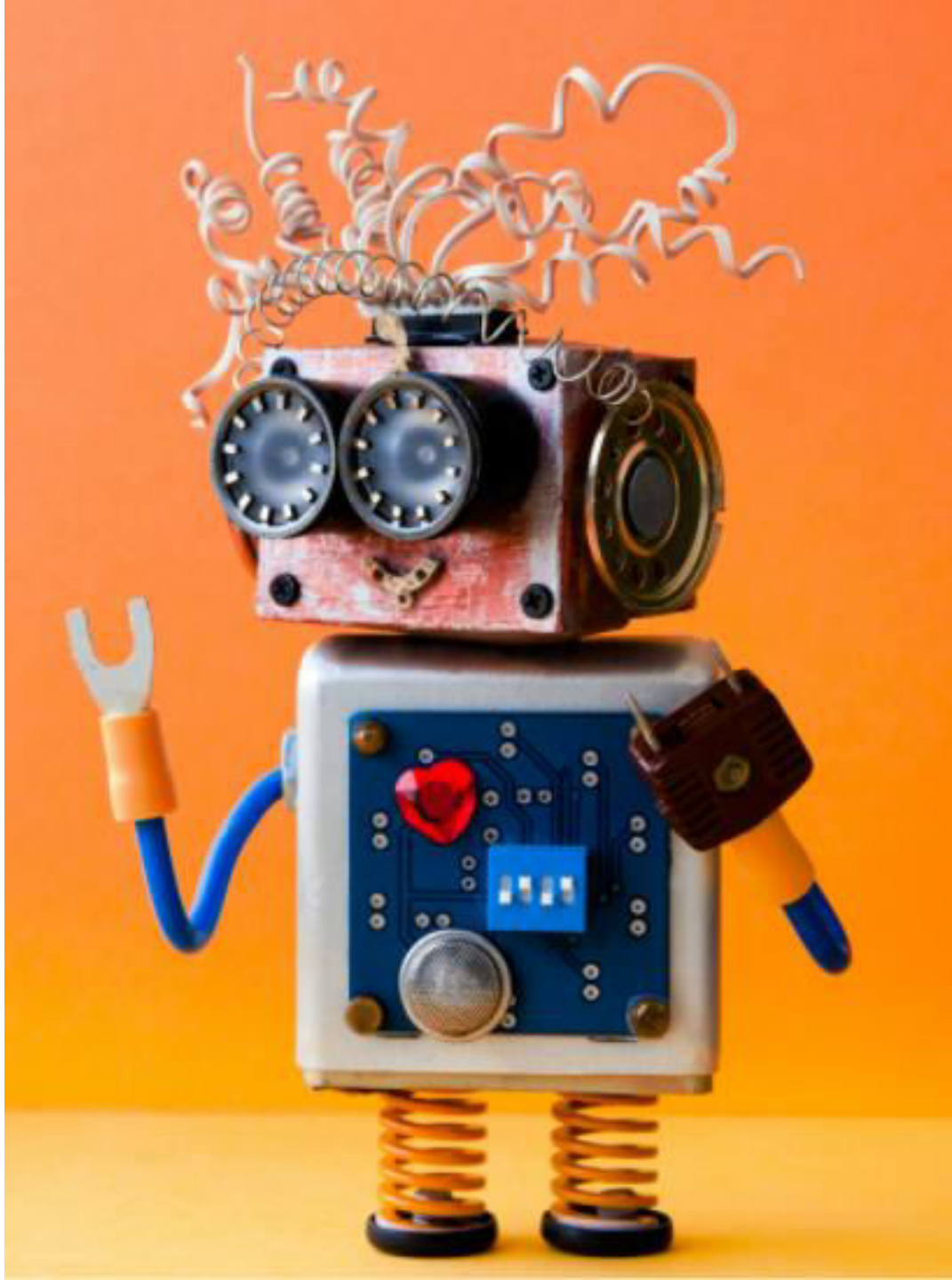
- Dataset:
  - Input (X): Outside Air Temperature
  - Output (Y): Overall daily revenue generated in dollars

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
5	14.352514	367.940744
6	13.707780	308.894518
7	30.833985	696.716640
8	0.976870	55.390338
9	31.669465	737.800824
10	11.455253	325.968408
11	3.664670	71.160153



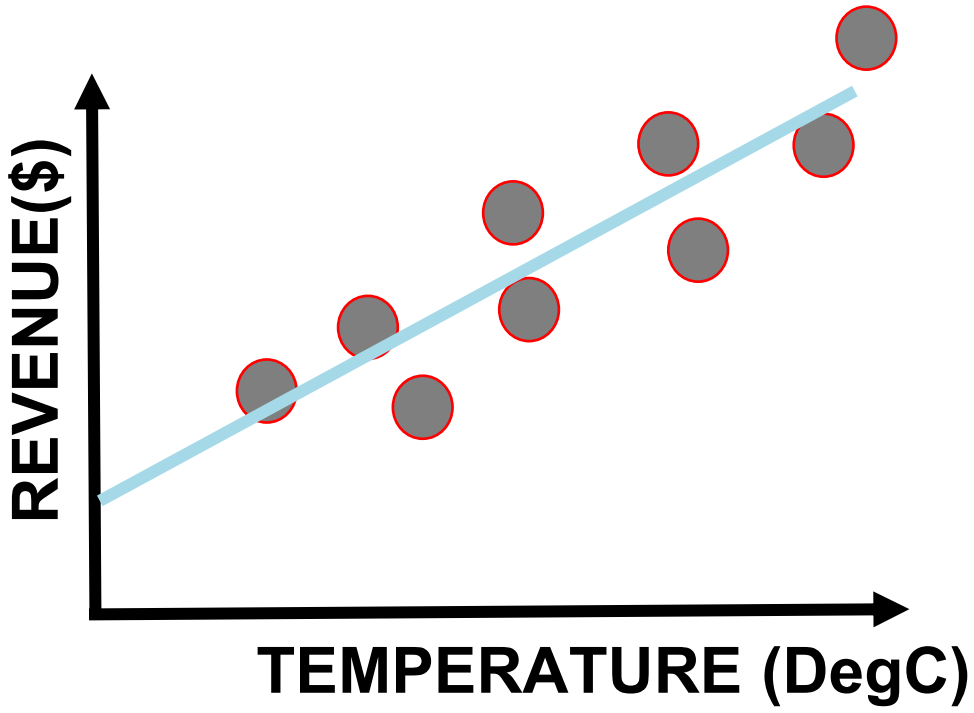
Source: https://www.goodfreephotos.com/vector-images/ice-cream-stand-vector-clipart.png.php

SIMPLE LINEAR REGRESSION 101



SIMPLE LINEAR REGRESSION 101

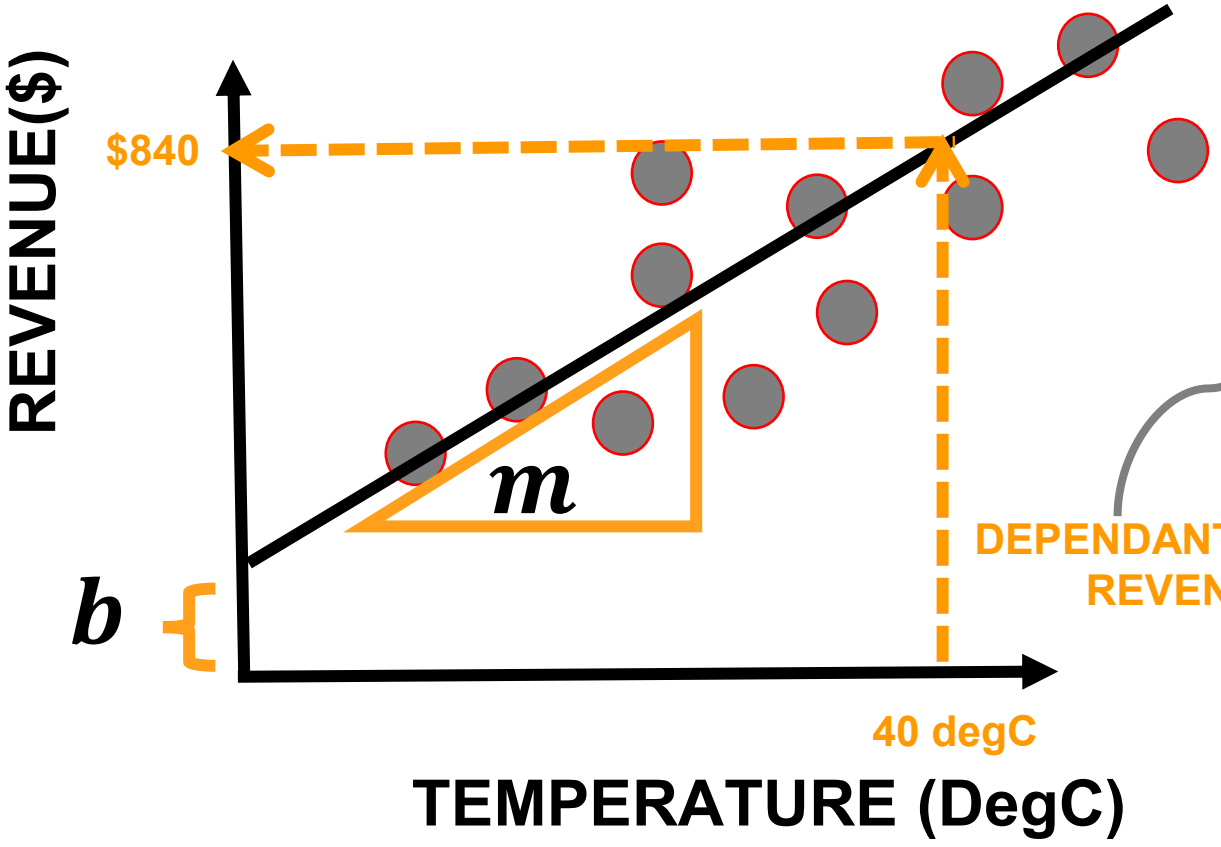
- In simple linear regression, we predict the value of one variable Y based on another variable X.
- X is called the independent variable and Y is called the dependant variable.
- Why simple? Because it examines relationship between two variables only.
- Why linear? when the independent variable increases (or decreases), the dependent variable increases (or decreases) in a linear fashion.



	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
5	14.352514	367.940744
6	13.707780	308.894518
7	30.833985	696.716640
8	0.976870	55.390338
9	31.669465	737.800824
10	11.455253	325.968408
11	3.664670	71.160153

SIMPLE LINEAR REGRESSION 101: SOME MATH!

- Goal is to obtain a relationship (model) between outside air temperature and ice cream sales revenue. Simply you need to find "*m*" and "*b*".
- This "trained" model can be later used to predict any Revenue (dollars) based on the outside air Temperature.



MODEL! (GOAL)

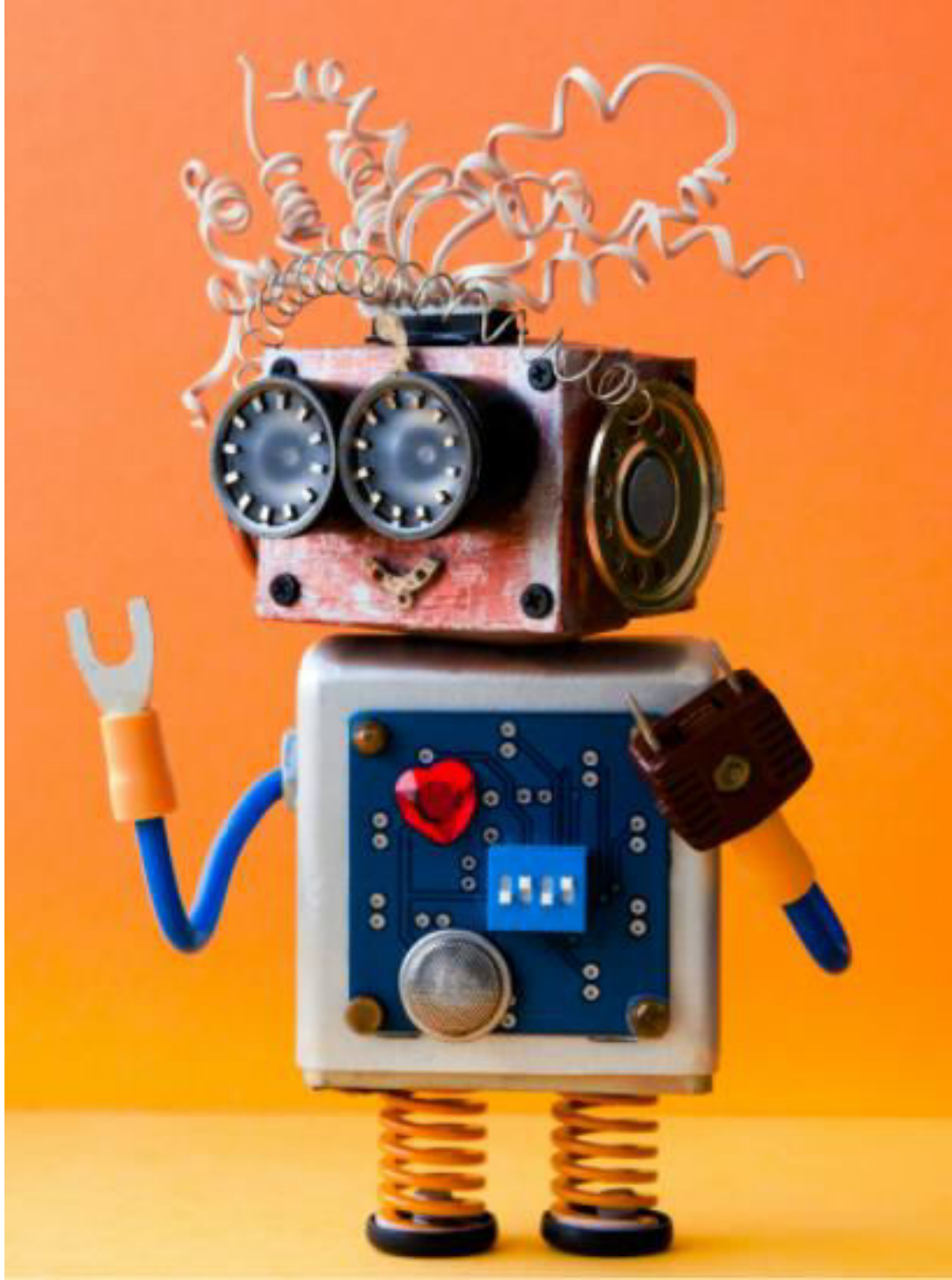
$$y = b + m * x$$

DEPENDANT VARIABLE
REVENUE ($)

INDEPENDENT VARIABLE
TEMPERATURE (DegC)

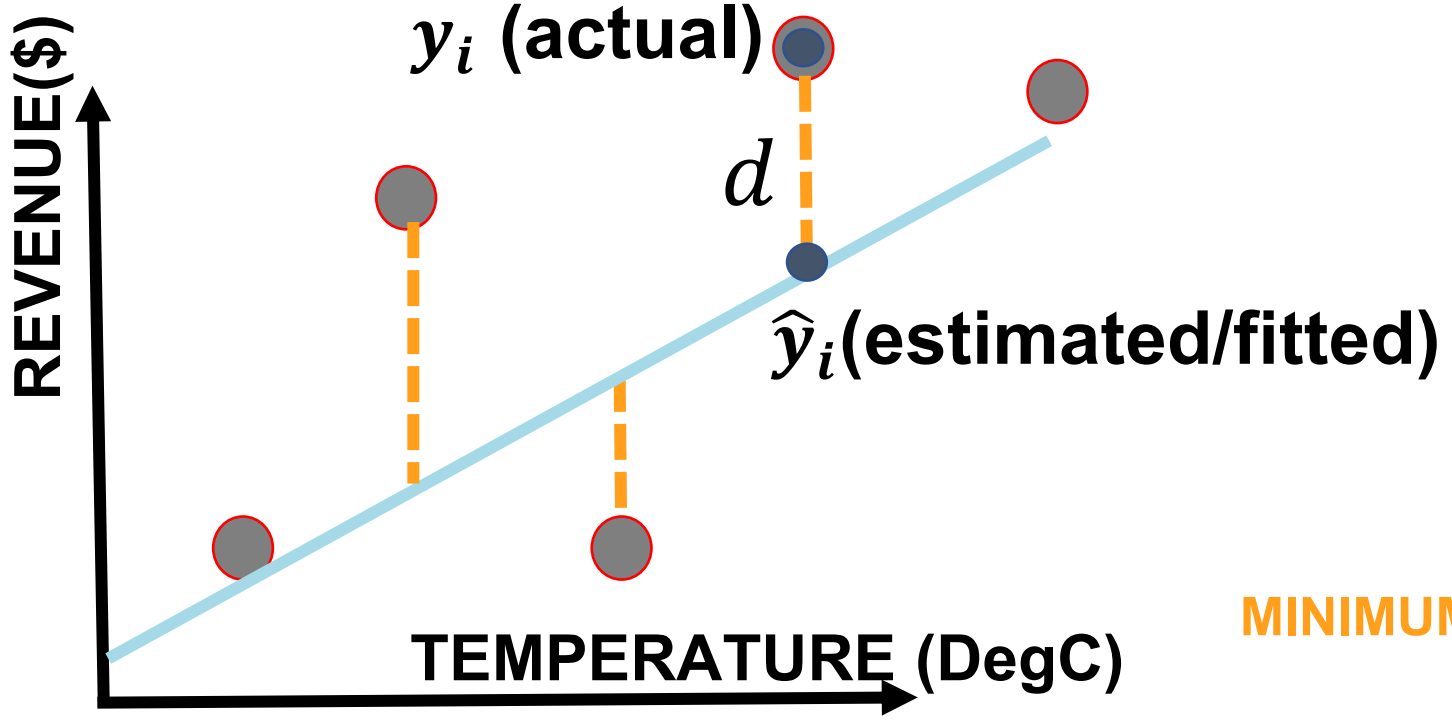
Temperature (x) = 40 degC
Revenue (y) = $840

LEAST SUM OF SQUARES



HOW TO GET MODEL PARAMETERS?
LEAST SUM OF SQUARES

- Least squares fitting is a way to find the best fit curve or line for a set of points.
- The sum of the squares of the offsets (residuals) are used to estimate the best fit curve or line.
- Least squares method is used to obtain the coefficients m and b.



$$d = \hat{y}_i - y_i$$

$$min \sum (\hat{y}_i - y_i)^2$$

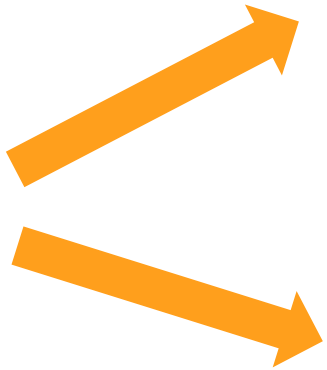
MINIMUM (LEAST) SUM OF SQUARES

TRAINING VS. TESTING DATASET

- Data set is divided into 75% for training and 25% for testing.
  - Training set: used for model training.
  - Testing set: used for testing trained model. Make sure that the testing dataset has never been seen by the trained model before.

TOTAL DATA

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
5	14.352514	367.940744
6	13.707780	308.894518
7	30.833985	696.716640
8	0.976870	55.390338
9	31.669465	737.800824
10	11.455253	325.968408
11	3.664670	71.160153

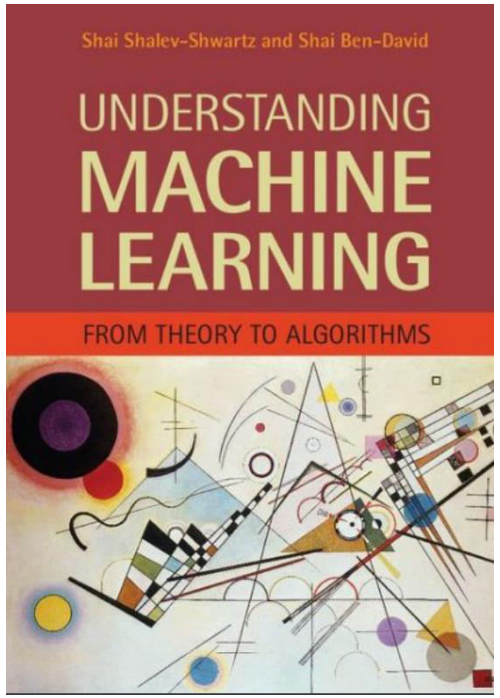


75% TRAINING

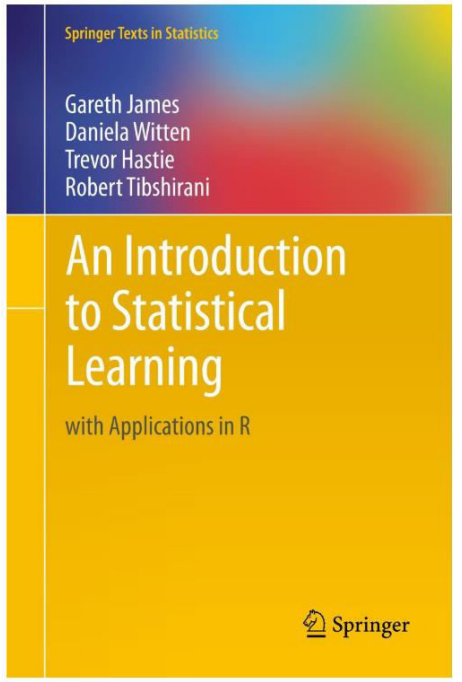
25% TESTING

SIMPLE LINEAR REGRESSION: ADDITIONAL READING MATERIAL

Additional Resources, Page #123:
http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf



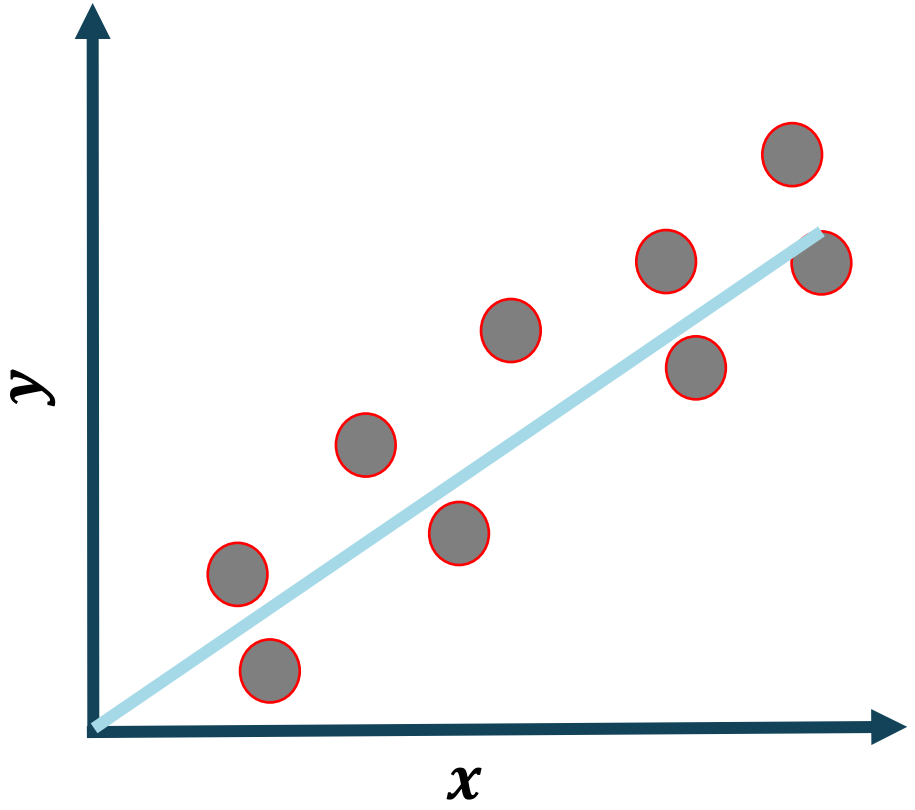
Additional Resources, Page #61:
http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf

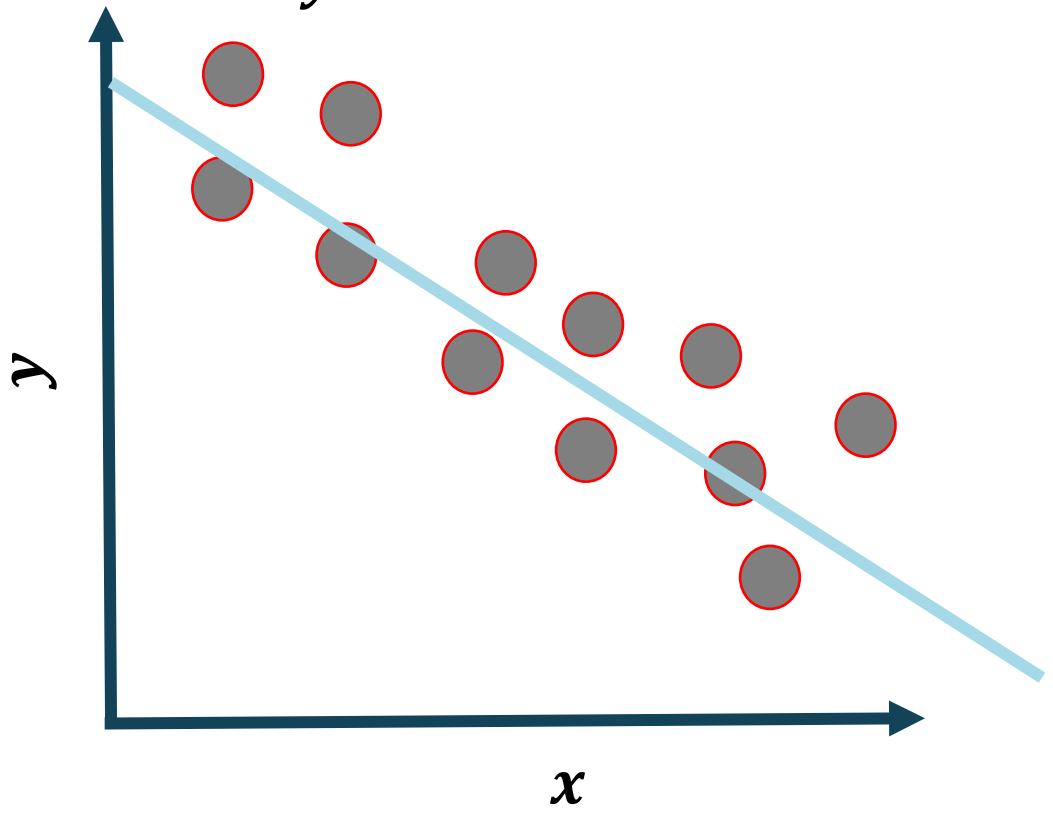


SIMPLE LINEAR REGRESSION: PRACTICE OPPORTUNITY

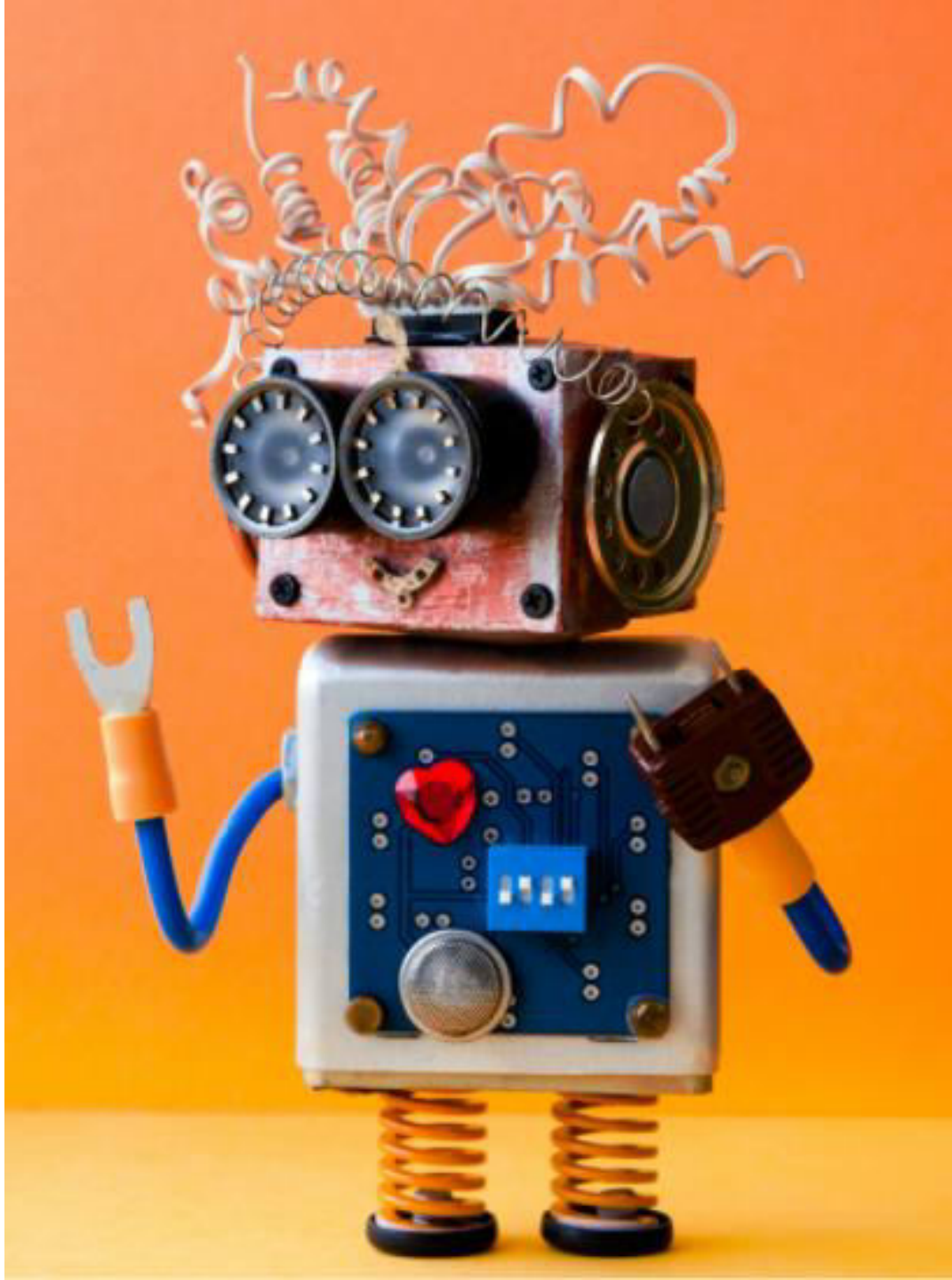
- Match the equations to the figures:

$y = 3 * x$ $y = 15 - 10 * x$



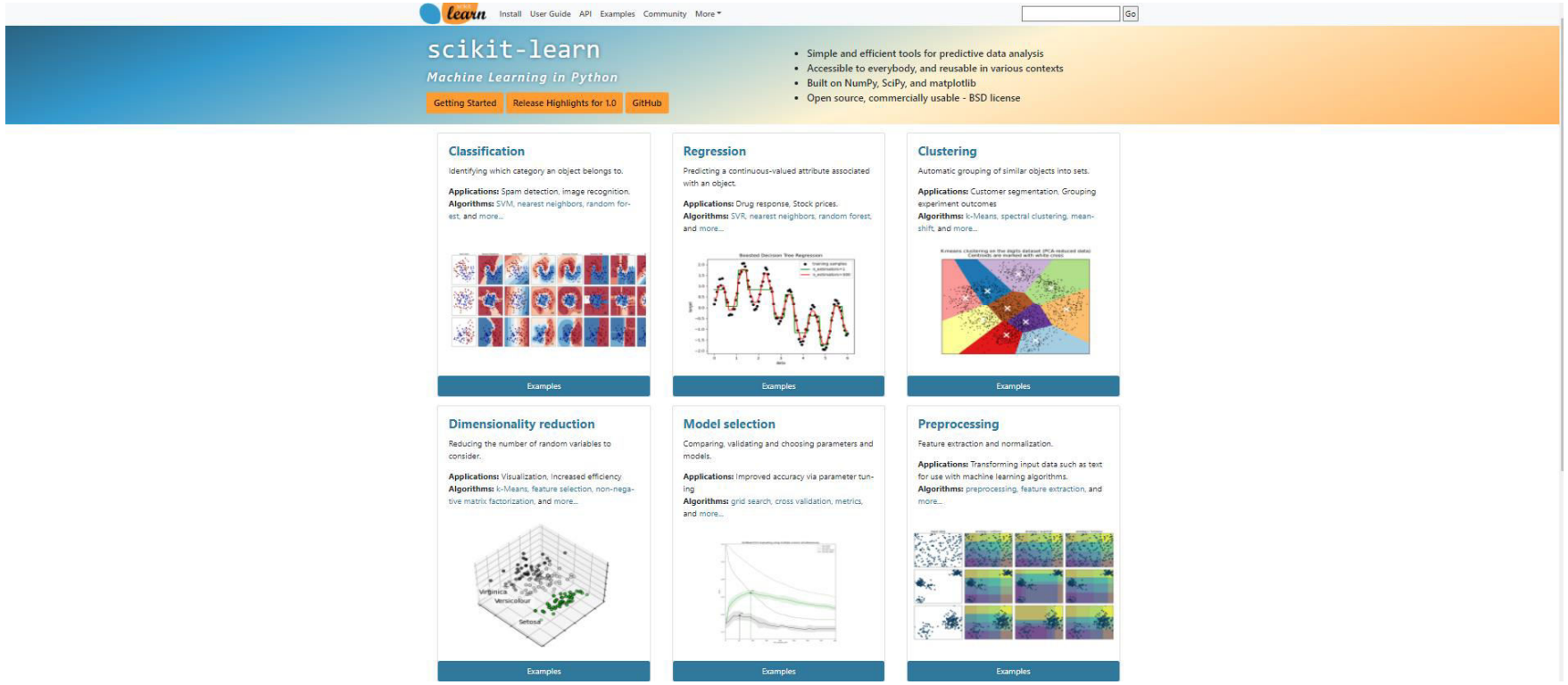


SCIKIT-LEARN 101



SCIKIT-LEARN 101

- Scikit-learn is a free machine learning library developed for python.
- Scikit-learn offers several algorithms for classification, regression, and clustering.
- Several famous models are included such as support vector machines, random forests, gradient boosting, and k-means.
- Scikit learn can be efficiently used in data preprocessing.



SCIKIT-LEARN 101: PERFORM DATA PRE-PROCESSING

- SCIKIT-Learn library is mostly used for data preprocessing such as standardization, scaling, normalization, and performing train test split.

```
# split the data into training and testing using SkLearn Library
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)
```

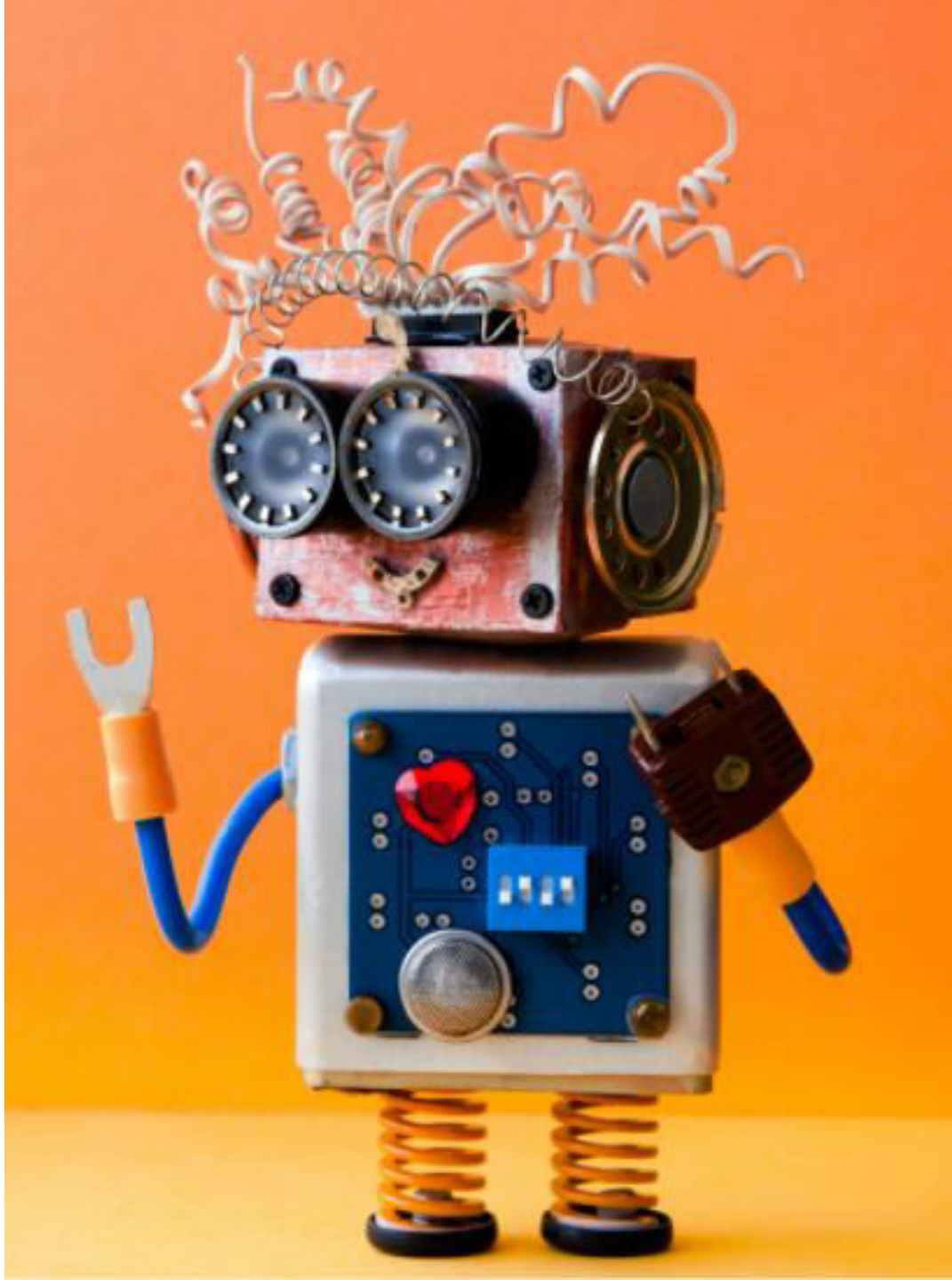
SCIKIT-LEARN 101: TRAIN A MACHINE LEARNING REGRESSION MODEL

- Check out the simple Linear Regression Examples in Scikit-Learn documentation:
- https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py

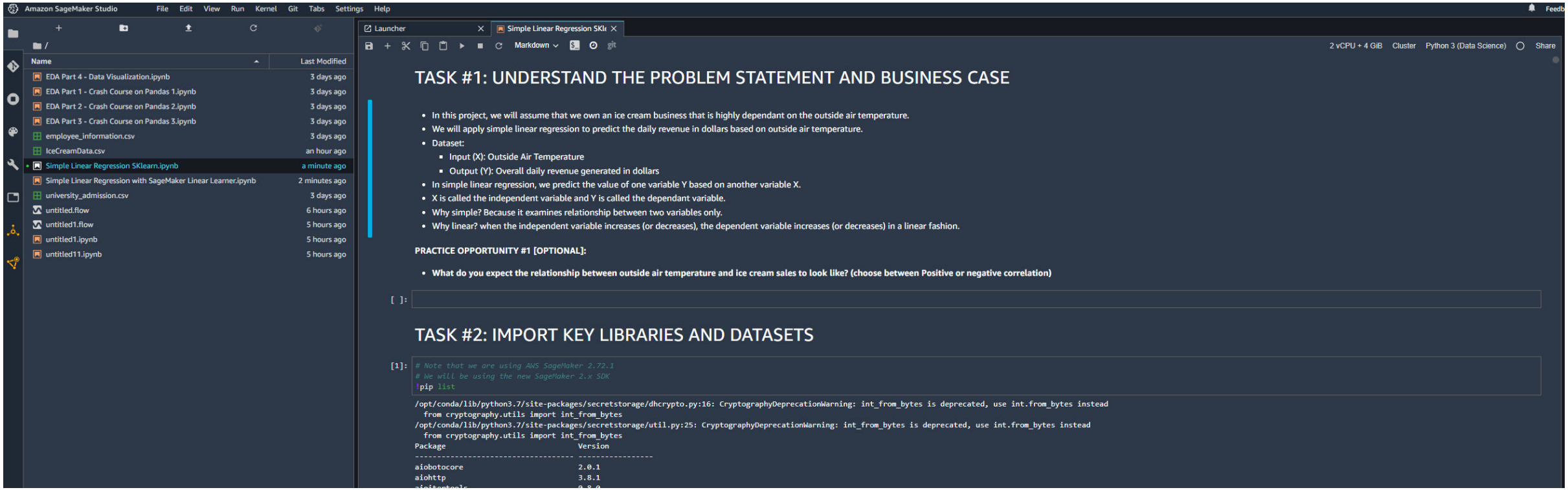
```
# using linear regression model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, accuracy_score

regresssion_model_sklearn = LinearRegression(fit_intercept = True)
regresssion_model_sklearn.fit(X_train, y_train)
```

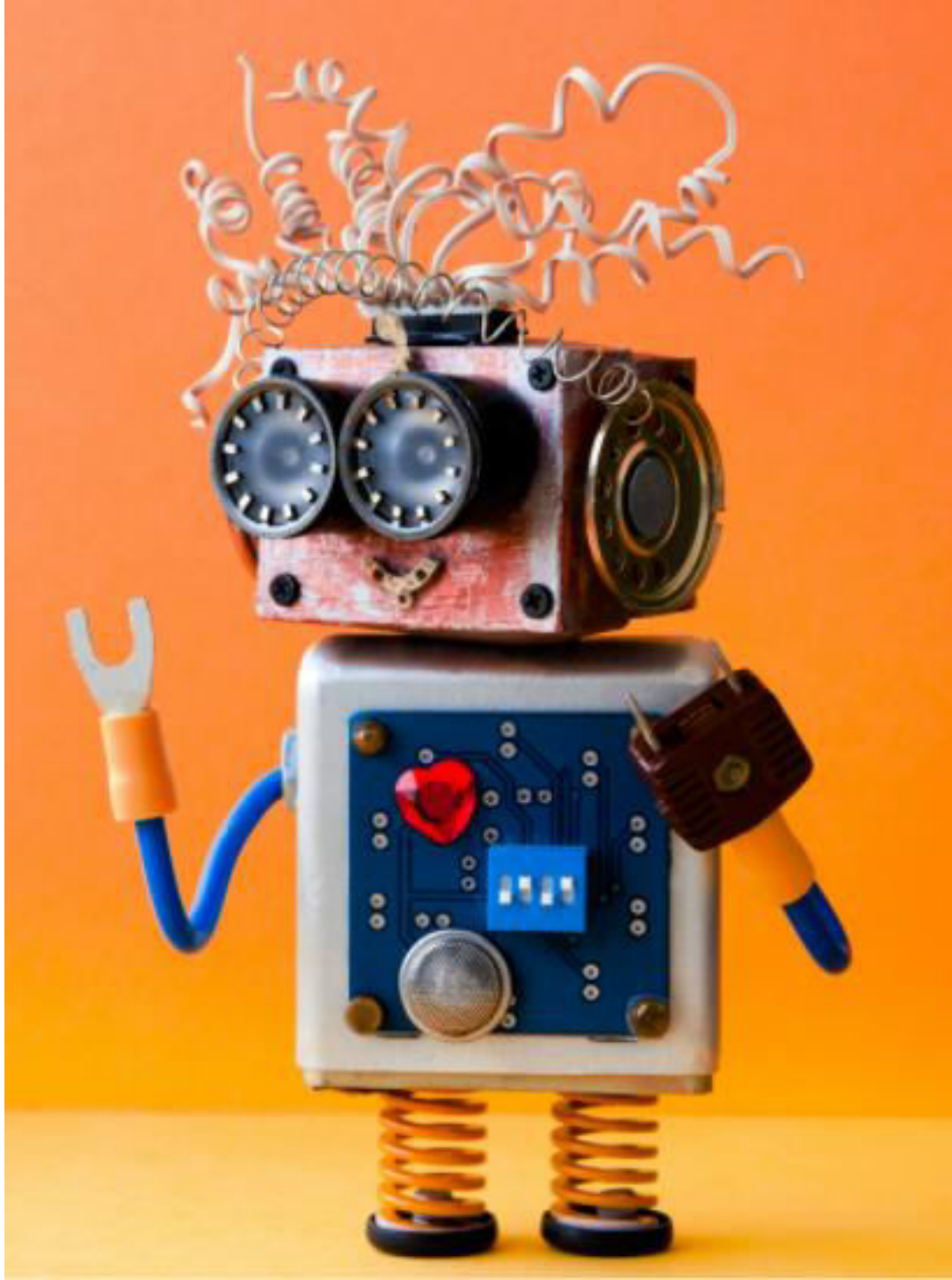
NOTEBOOK CODE DEMO



NOTEBOOK DEMO



END-OF-DAY PROJECT



END-OF-DAY PROJECT

- You have been hired as a consultant to a major Automotive Manufacturer and you have been tasked to develop a model to predict the impact of increasing the vehicle horsepower (HP) on fuel economy (Mileage Per Gallon (MPG)). You gathered the following dataset:
    - Independent variable X: Vehicle Horsepower
    - Dependent variable Y: Mileage Per Gallon (MPG)

Horse Power	Fuel Economy (MPG)
118.7707988	29.34419493
176.3265674	24.6959341
219.2624649	23.95201001
187.3100089	23.38454579
218.5943396	23.42673926
175.8381062	24.17357106
271.4416078	17.16358348
294.4259159	17.27421781
126.2110081	28.71821022
163.3503346	28.28951641
321.840752	17.30062804
120.4842359	29.67863744
155.4153676	27.29492955
191.7148134	23.55672887
211.7291092	25.34189228
259.1831915	20.46737357
236.5717375	23.18528033
191.0989631	24.98962965
123.8856983	29.3933298
136.3064532	31.49742937
212.7389563	23.20474499
232.4499479	22.3130506
122.0401613	31.79661213

END-OF-DAY PROJECT

Using the skeleton jupyter notebook "*Simple Linear Regression - Fuel Consumption Project Questions*", perform the following:

- 1. Load the "*FuelEconomy.csv*" dataset
- 2. Perform data visualization and basic exploratory data analysis
- 3. Split the data into 80% for training and 20% for testing
- 4. Train a machine linear regression model in Scikit-Learn
- 5. Assess trained model performance