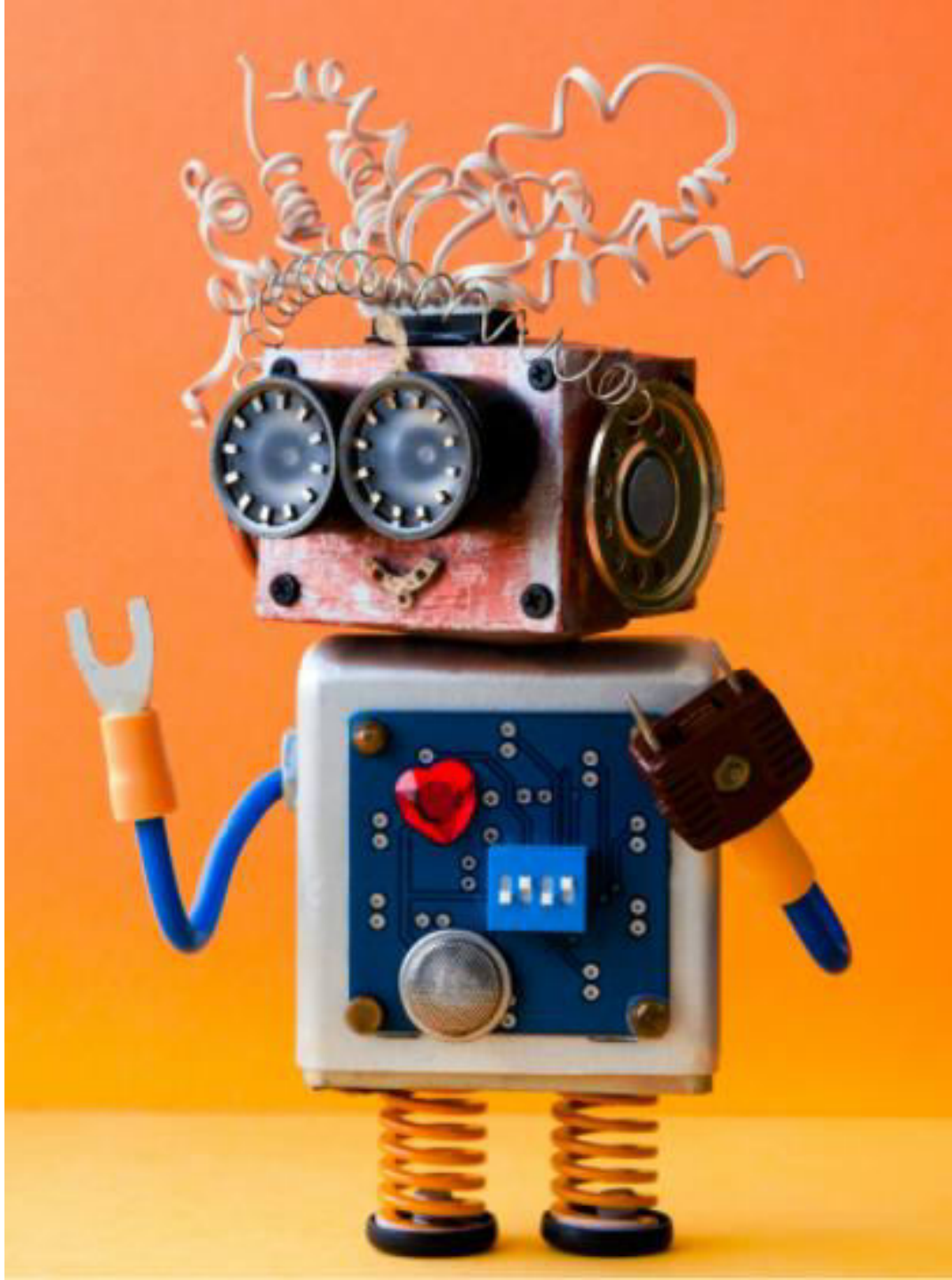
OVERVIEW OF AWS SAGEMAKER BUILT-IN ALGORITHMS



SAGEMAKER AVAILABLE ALGORITHMS

CLASSIFICATION

Linear Learner
XG-Boost
K-Nearest Neighbors (KNN)
Factorization Machines

REGRESSION

Linear Learner
XG-Boost
K-Nearest Neighbors (KNN)

COMPUTER VISION

Image Classification
Object Detection
Semantic Segmentation

DIMENSIONALITY REDUCTION

Principal Component Analysis (PCA)
Object2Vec

TIME SERIES FORECASTING

DeepAR

RECOMMENDATION

Factorization Machines

TEXT AND TOPIC MODELING

Blazing Text
Neural Topic Modelling (NTM)
Latent Dirichlet Allocation (LDA)
Sequence2Sequence

ANOMALY DETECTION

Random Cut Forest
IP Insights

CLUSTERING

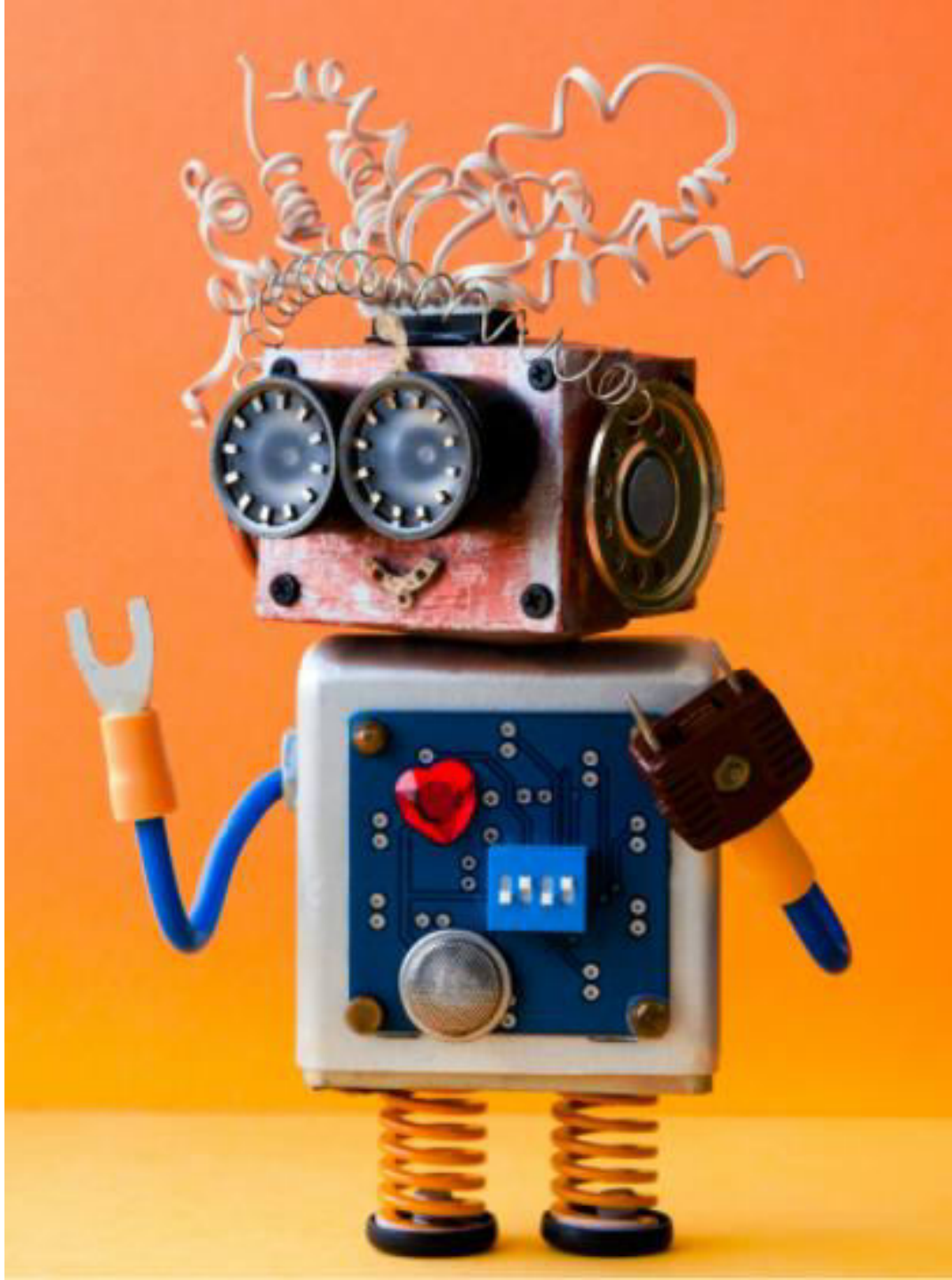
K-Means
KNN

SAGEMAKER AVAILABLE ALGORITHMS

BlazingText Word2Vec	BlazingText implementation of the Word2Vec algorithm for scaling and accelerating the generation of word embeddings from a large number of documents.
DeepAR	An algorithm that generates accurate forecasts by learning patterns from many related time-series using recurrent neural networks (RNN).
Factorization Machines	A model with the ability to estimate all of the interactions between features even with a very small amount of data.
Gradient Boosted Trees (XGBoost)	Short for “Extreme Gradient Boosting”, XGBoost is an optimized distributed gradient boosting library.
Image Classification (ResNet)	A popular neural network for developing image classification systems.
IP Insights	An algorithm to detect malicious users or learn to usage patterns of IP addresses.
K-Means Clustering	One of the simplest ML algorithms. It’s used to find groups within unlabeled data.
K-Nearest Neighbor (k-NN)	An index based algorithm to address classification and regression based problems.
Latent Dirichlet Allocation (LDA)	A model that is well suited to automatically discovering the main topics present in a set of text files.
Linear Learner (Classification)	Linear classification uses an object’s characteristics to identify the appropriate group that it belongs to.
Linear Learner (Regression)	Linear regression is used to predict the linear relationship between two variables.
Neural Topic Modelling (NTM)	A neural network based approach for learning topics from text and image datasets.
Object2Vec	A neural-embedding algorithm to compute nearest neighbors and to visualize natural clusters.
Object Detection	Detects, classifies, and places bounding boxes around multiple objects in an image.
Principal Component Analysis (PCA)	Often used in data pre-processing, this algorithm takes a table or matrix of many features and reduces it to a smaller number of representative features.
Random Cut Forest	An unsupervised machine learning algorithm for anomaly detection.
Semantic Segmentation	Partitions an image to identify places of interest by assigning a label to the individual pixels of the image.
Seqence2Sequence	A general-purpose encoder-decoder for text that is often used for machine translation, text summarization, etc.

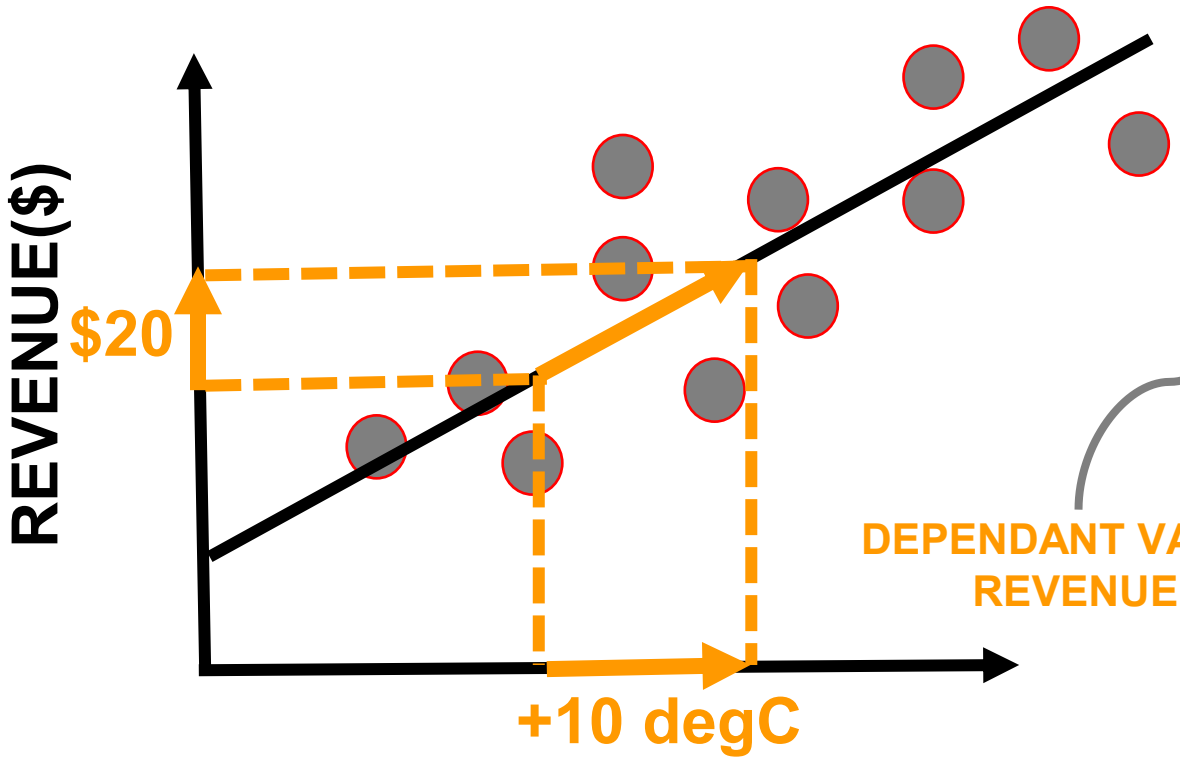
Source: https://aws.amazon.com/sagemaker/build/

LINEAR LEARNER IN SAGEMAKER



SIMPLE LINEAR REGRESSION 101: SOME MATH!

- 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.
- Goal is to obtain a relationship (model) between outside air temperature and ice cream sales revenue



MODEL! (GOAL)

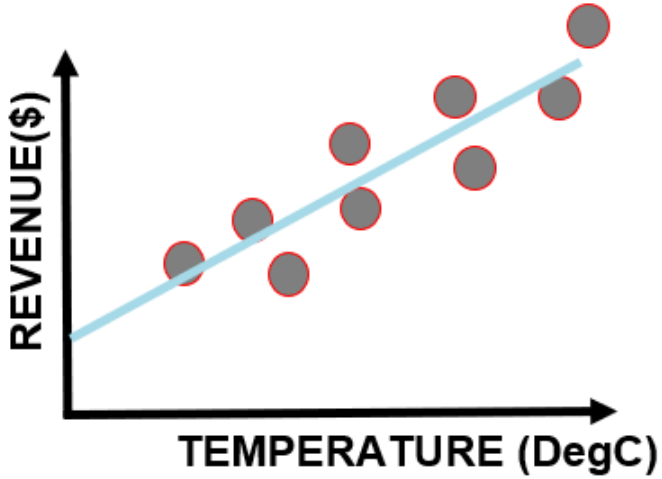
$$y = b + m * x$$

DEPENDANT VARIABLE
REVENUE ($)

INDEPENDENT VARIABLE
TEMPERATURE (DegC)

SAGEMAKER LINEAR LEARNER: OVERVIEW

- Linear Learner is a supervised learning algorithm that is used to fit a line to the training data.
- It could be used for both classification and regression tasks as follows:
  - **Regression:** output contains continuous numeric values
  - **Binary classification:** output label must be either 0 or 1 (linear threshold function is used).
  - **Multiclass classification:** output labels must be from 0 to *num_classes - 1*.
- The best model optimizes either of the following:
  - **For regression:** focus on Continuous metrics such as mean square error, root mean squared error, cross entropy loss, absolute error.
  - **For classification:** focus on discrete metrics such as F1 score, precision, recall, or accuracy.



	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

SAGEMAKER LINEAR LEARNER: USE CASES

REGRESSION TASKS

- Revenue predictions based on previous years performance.

DISCRETE BINARY CLASSIFICATION

- Does this patient have a disease or not?

DISCRETE MULTICLASS CLASSIFICATION

- Should an autonomous car stop, slow down or accelerate?

SAGEMAKER LINEAR LEARNER: OVERVIEW

Preprocessing

- Normalization or feature scaling is offered by Linear Learner (which is great!)
- Feature scaling is a critical preprocessing step to ensure that the model does not become dominated by the weight of a single feature.

Training

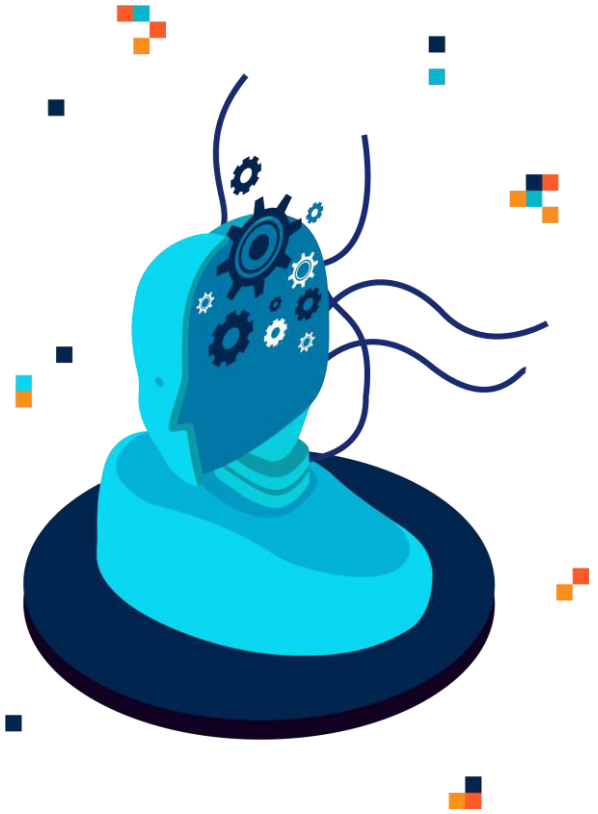
- Linear Learner uses stochastic gradient descent to perform the training
- Select an appropriate optimization algorithm such as Adam, AdaGrad, and SGD
- Hyperparameters can be selected and tuned (Example: learning rate).
- Overcome model overfitting using L1, L2 regularization

Validation

- Trained models are evaluated against a validation dataset and best model is selected based on the following metrics:
    - For regression: mean square error, root mean squared error, cross entropy loss, absolute error.
    - For classification: F1 score, precision, recall, or accuracy.

SAGEMAKER LINEAR LEARNER HYPERPARAMETERS

- **Predictor_type:** ‘regressor’
- **Learning Rate:** The step size used by the optimizer for parameter updates.
- **L1:** L1 regularization parameter.
- **Mini_batch_size:** The number of observations per mini-batch
- **Wd:** The weight decay parameter, also known as the L2 regularization parameter.
- Check out the rest of hyperparameters here: https://docs.aws.amazon.com/sagemaker/latest/dg/ll_hyperparameters.html

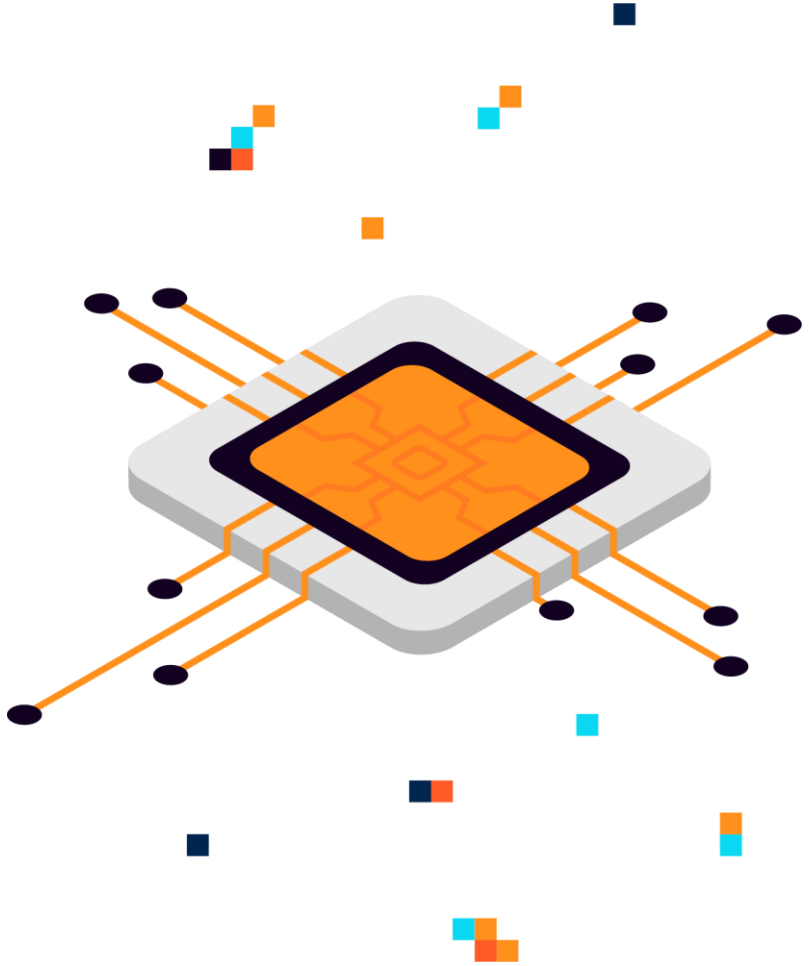


SAGEMAKER LINEAR LEARNER: INPUT/OUTPUT DATA

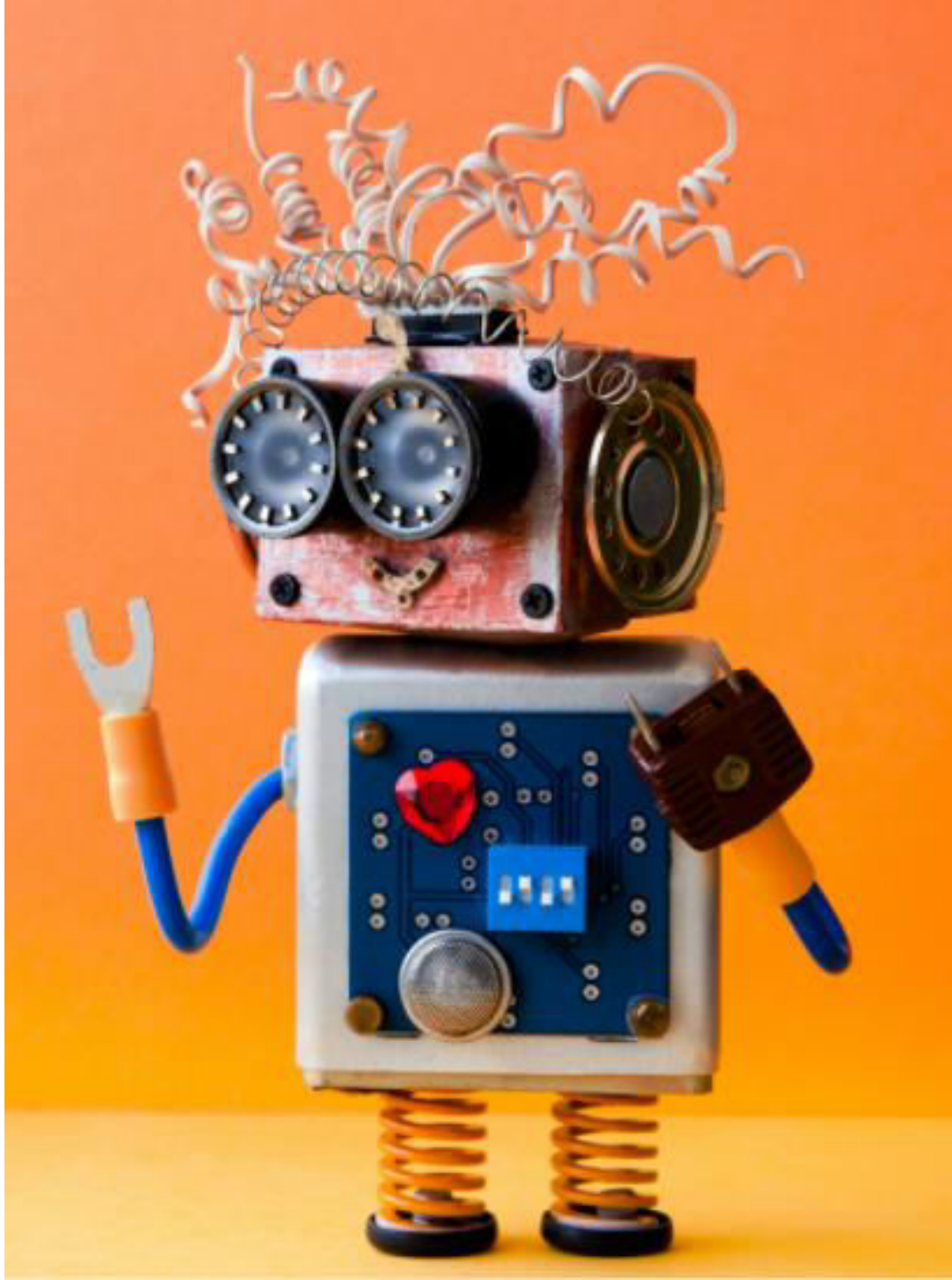
- Amazon SageMaker linear learner supports the following input data types:
  - RecordIO-wrapped protobuf (note: only Float32 tensors are supported)
  - Text/CSV (note: First column assumed to be the target label)
  - File or Pipe mode both supported
- For inference, linear learner algorithm supports the application/json, application/x-recordio-protobuf, and text/csv formats.
- For regression (predictor_type='regressor'), the score is the prediction produced by the model.

SAGEMAKER LINEAR LEARNER: EC2 INSTANCE

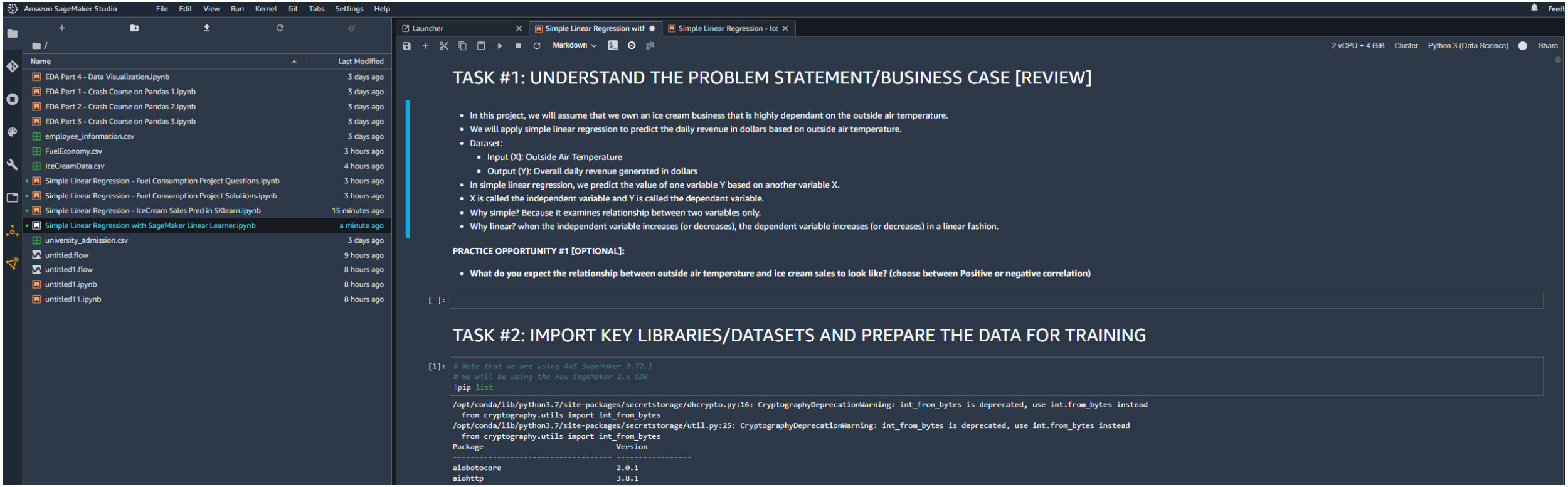
- Linear Learner algorithm could be trained on:
  - Single CPU and GPU instances
  - Multi-Machine CPU and GPU instances
- During testing, multi-GPU computers are not necessary (add cost with no value).



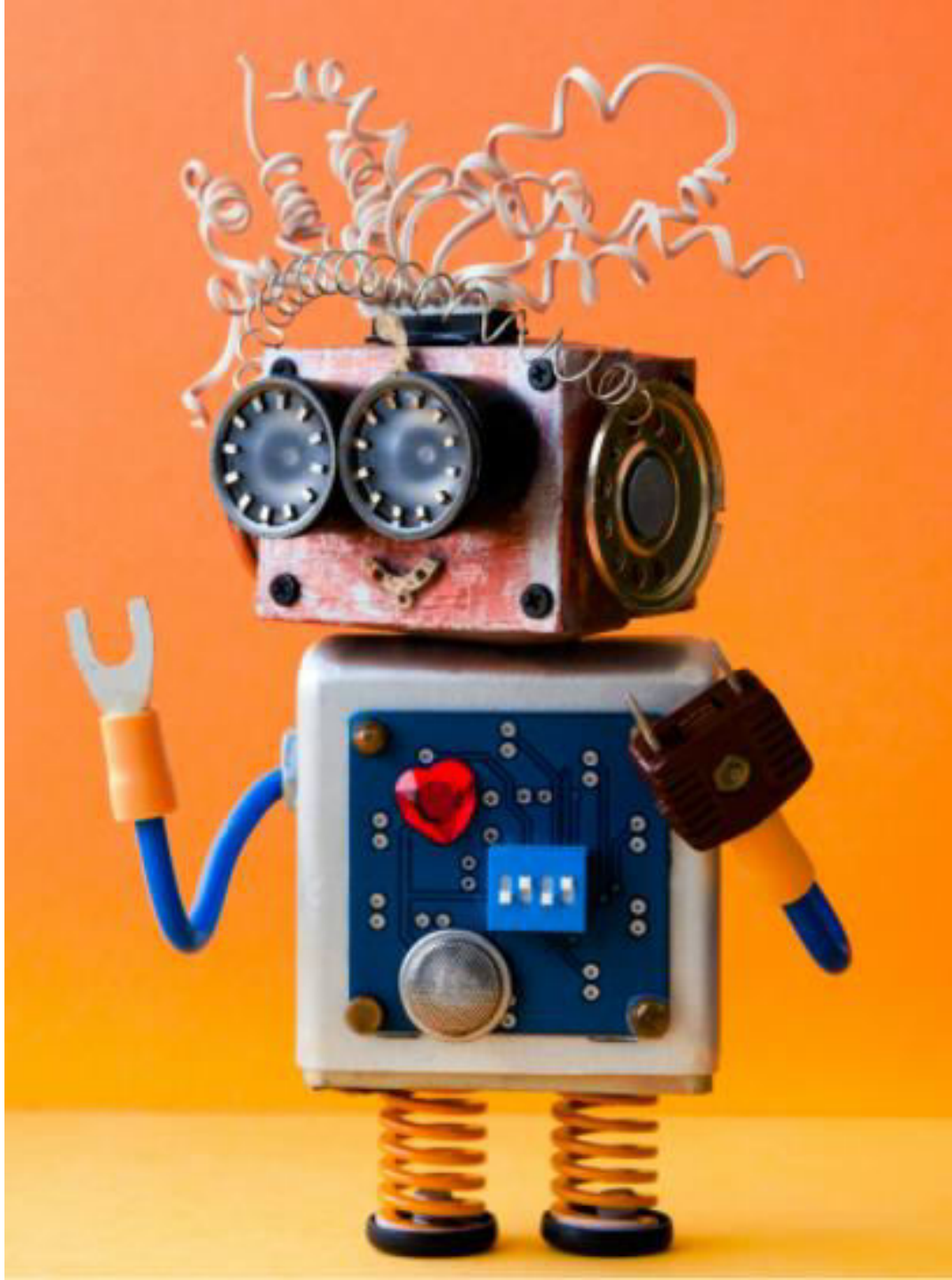
LINEAR LEARNER MODEL TRAINING DEMO



LINEAR LEARNER MODEL TRAINING DEMO:



END-OF-DAY FINAL PROJECT



PROJECT OVERVIEW

- 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

PROJECT TASKS

Using AWS SageMaker, perform the following:

- 1. Load the "*FuelEconomy.csv*" dataset to S3
- 2. Split the data into 80% for training and 20% for testing
- 3. Train a linear learner regression model using SageMaker SDK
- 4. Deploy trained model as an endpoint
- 5. Assess trained model performance, what is R2?
- 6. Delete the endpoint