哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Big data technology and Practice

李春山

2020年9月22日

内容提要

Chapter 5: Machine Learning – Overview

Chapter 6: ML Using Linear Regression

Chapter 5: Machine Learning - Overview

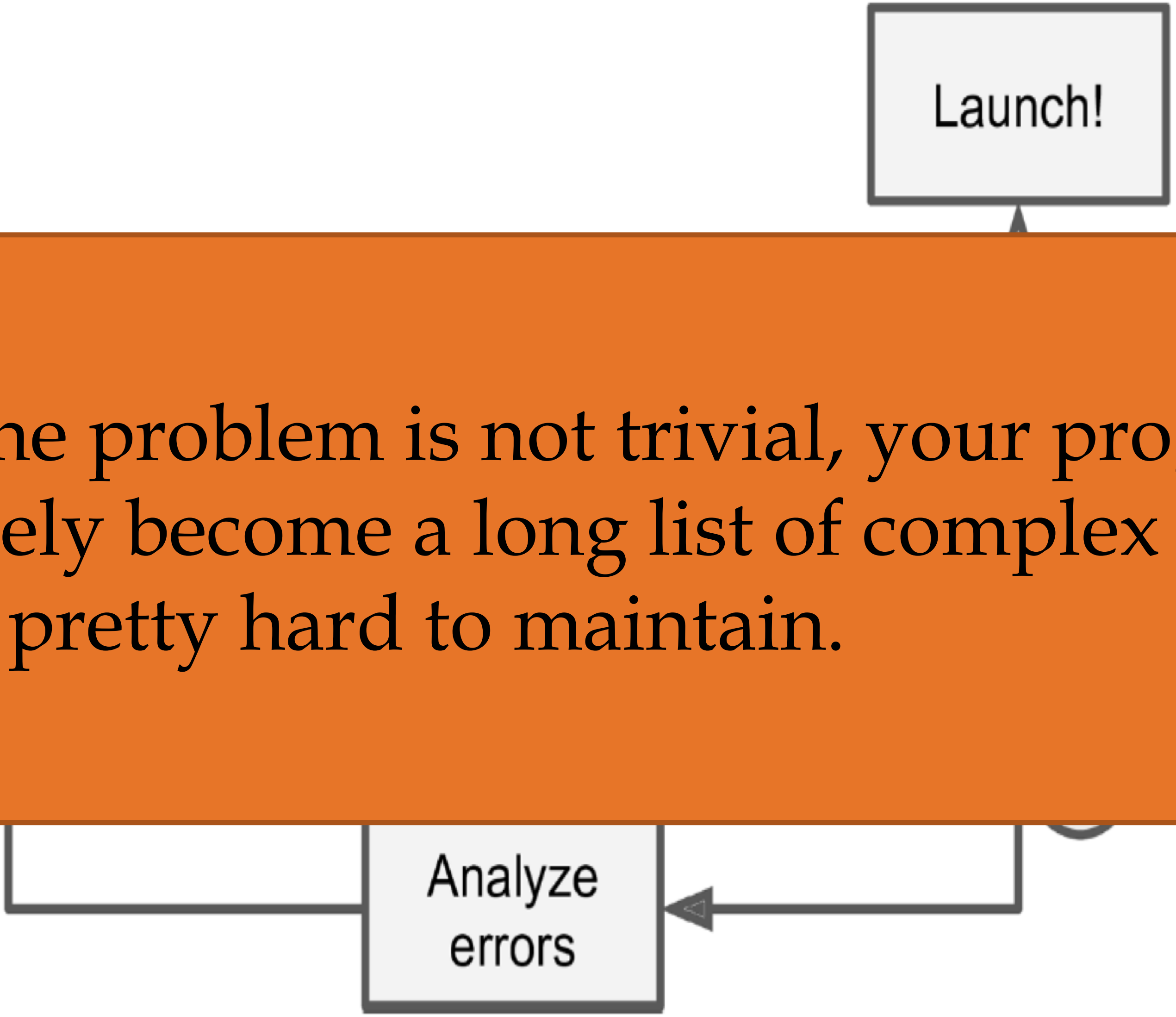
- **What is Machine Learning (ML)?**
- **There are different ways to define "ML" from different point of view.**
- **In this subject, the definition:**
  - "Machine learning is a method of data analysis that automates analytical model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look."

2. Why Use Machine Learning?

- Consider an example: how to write a program to filter a spam e-mail,

- See the difference between traditional way from ML method

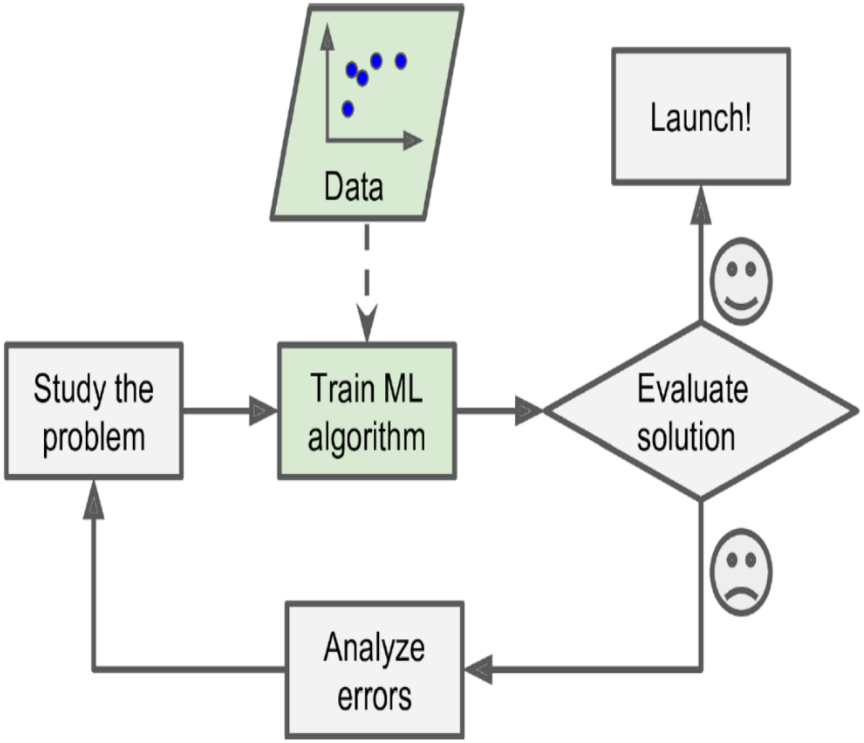
Traditional way

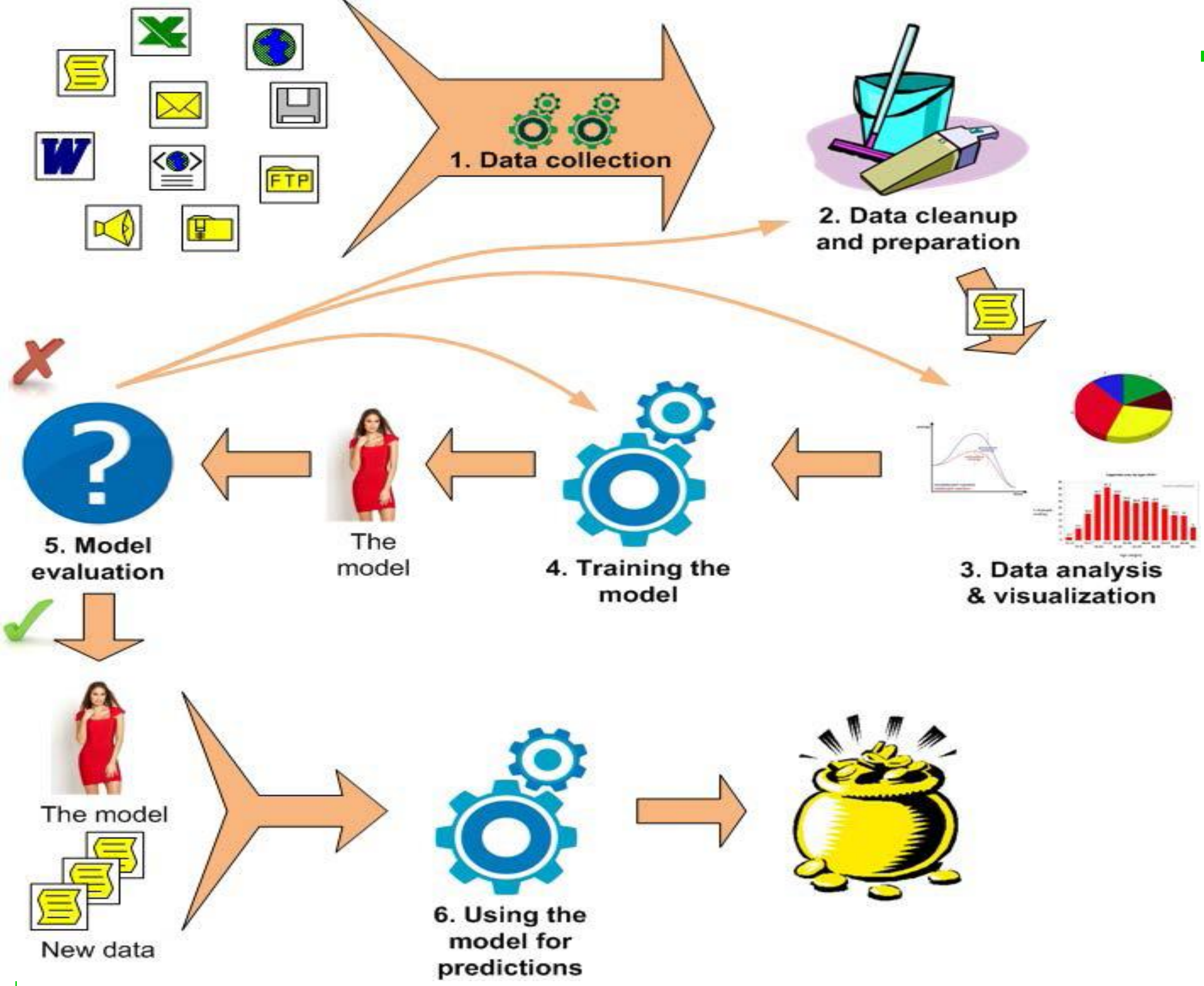


Since the problem is not trivial, your program will likely become a long list of complex rules—pretty hard to maintain.

Using Machine Learning (ML)

- In contrast, a spam filter based on ML can automatically learn which words and phrases are good predictors of spam.
- by detecting unusually frequent patterns of words in the spam examples compared to the ham examples.





1. Data collection

2. Data cleanup and preparation

3. Data analysis & visualization

4. Training the model

The model

5. Model evaluation

The model

New data

6. Using the model for predictions

Machine Learning is great for:

- Problems for which existing solutions require a lot of hand-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better.

- Complex problems for which there is no good solution at all using a traditional approach: the best Machine Learning techniques can find a solution.

Machine Learning is great for:

- Fluctuating environments: a Machine Learning system can adapt to new data.

- Getting insights about complex problems and large amounts of data

What is used ML for

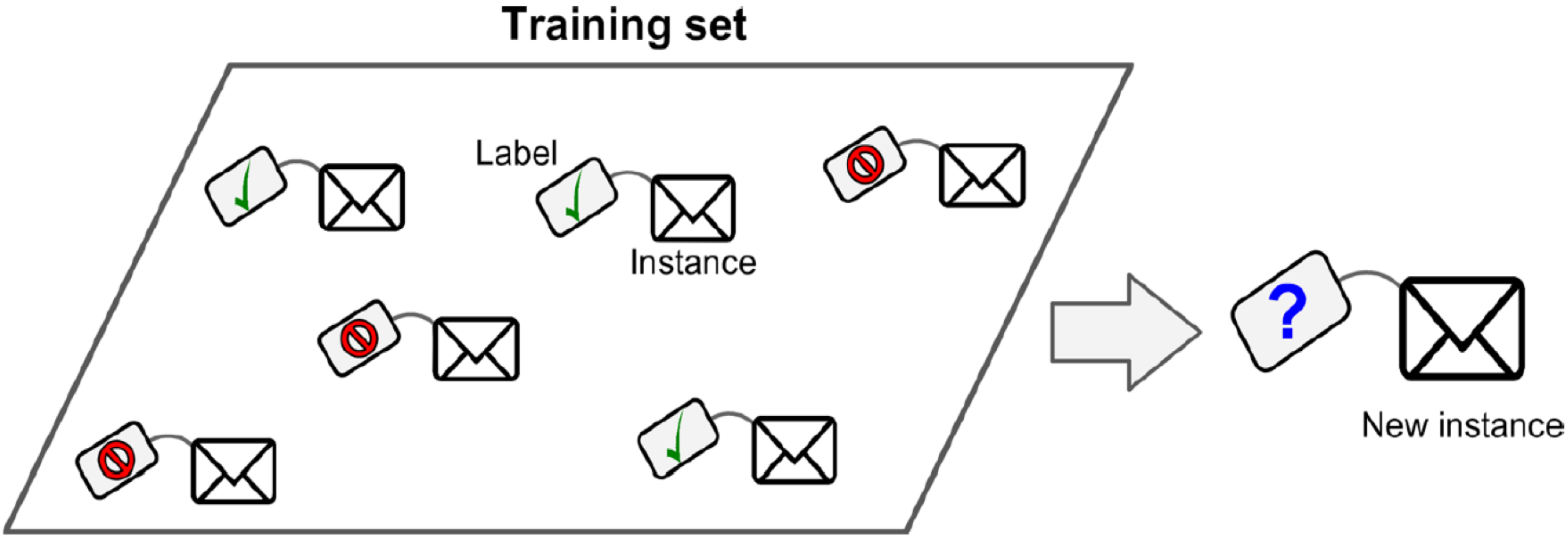
- Fraud detection.
- Web search results.
- Real-time ads on web pages
- Credit scoring and next-best offers.
- Prediction of equipment failures.
- New pricing models.

What is used ML for

- Network intrusion detection.
- Recommendation Engines
- Customer Segmentation
- Text Sentiment Analysis
- Predicting Customer Churn
- Pattern and image recognition.
- Email spam filtering.
- Financial Modeling

Types of Machine Learning Systems

- **1). Supervised and Unsupervised Learning**
  - In supervised learning, the training data you feed to the algorithm includes the desired solutions, called labels.



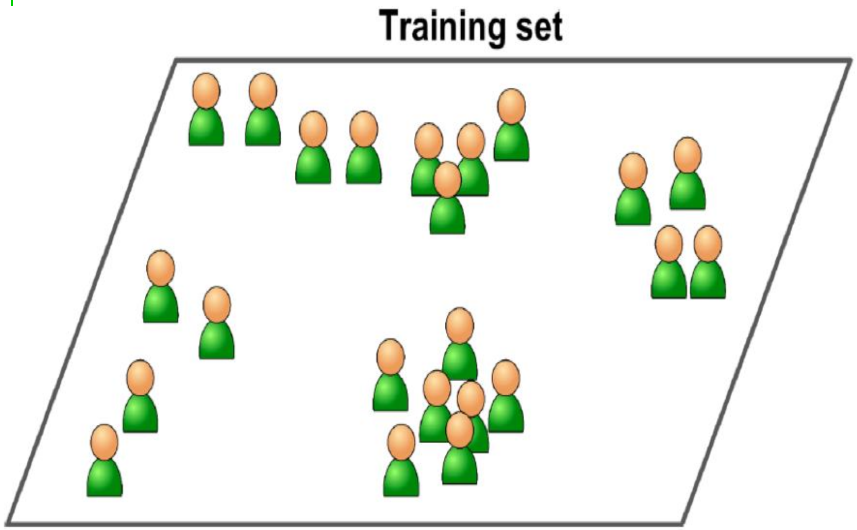
Types of Machine Learning Systems

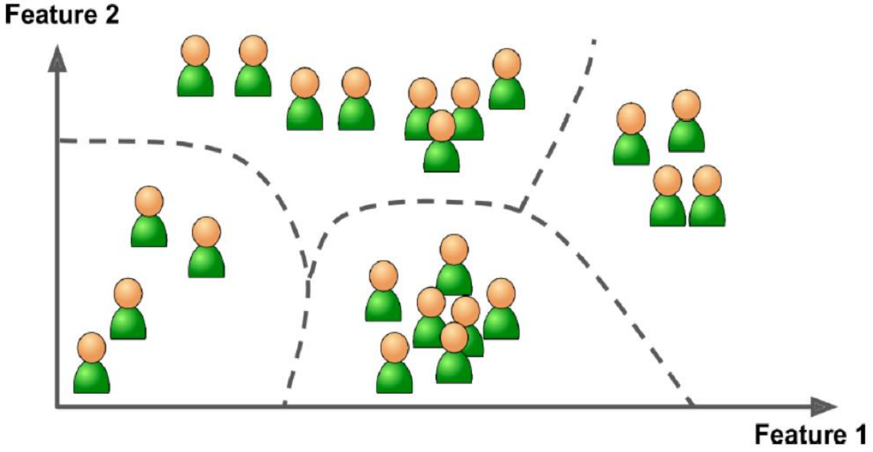
- **A typical supervised learning task is classification, such as, a spam filter with two class (spam or ham); it is trained with many e-mails; it must learn how to classify new emails.**

- **Another is to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.). To train the system, need many examples of cars, including both their predictors and their labels (i.e., their prices).**

Types of Machine Learning Systems

- In unsupervised learning, as you might guess, the training data is unlabeled.
- The system tries to learn without a teacher.

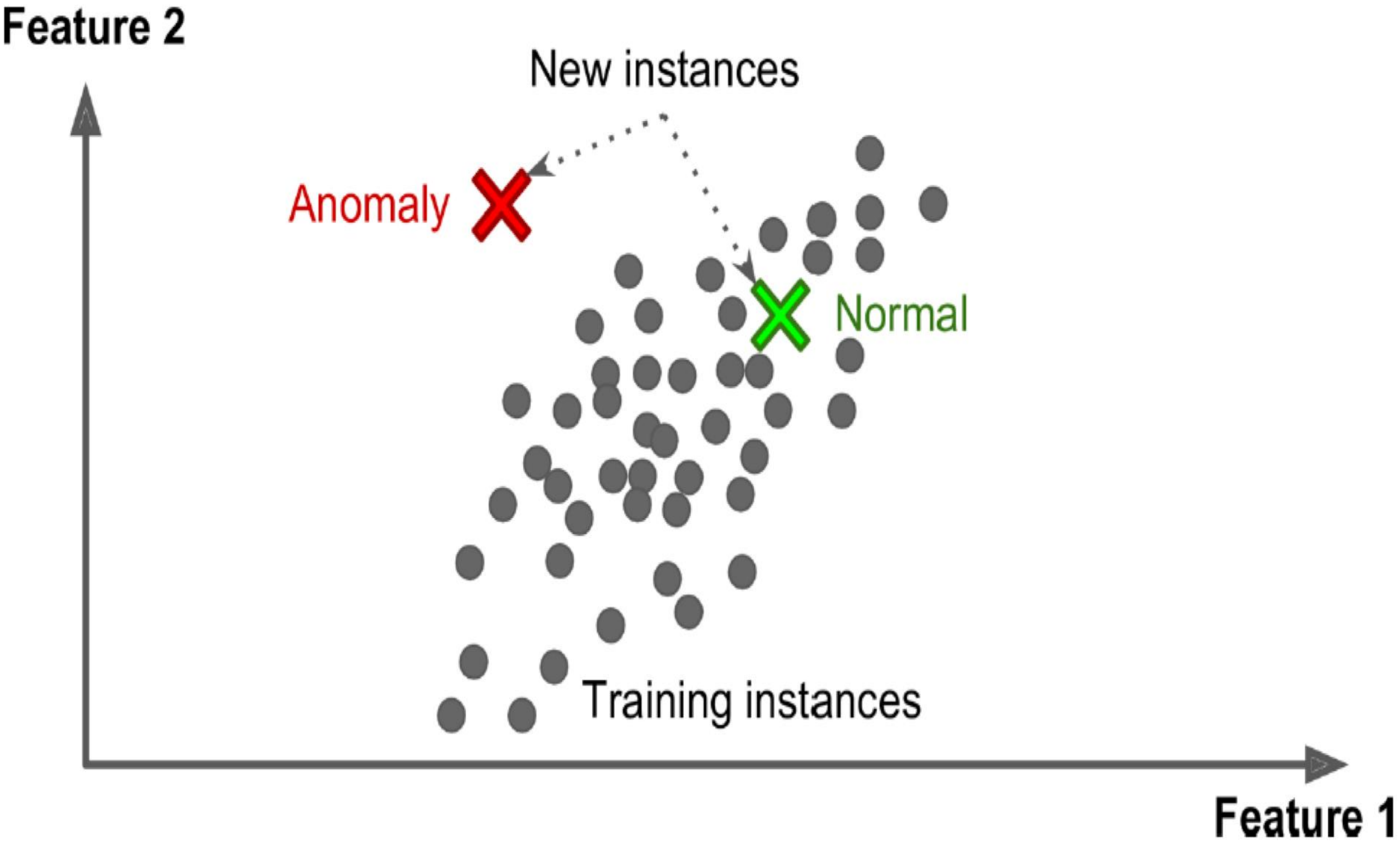




Types of Machine Learning Systems

- For example, from a lot of data about your blog's visitors to run a ML clustering algorithm to try to detect groups of similar visitors.
- No information about which group a visitor belongs to.
- it finds those connections without your help.

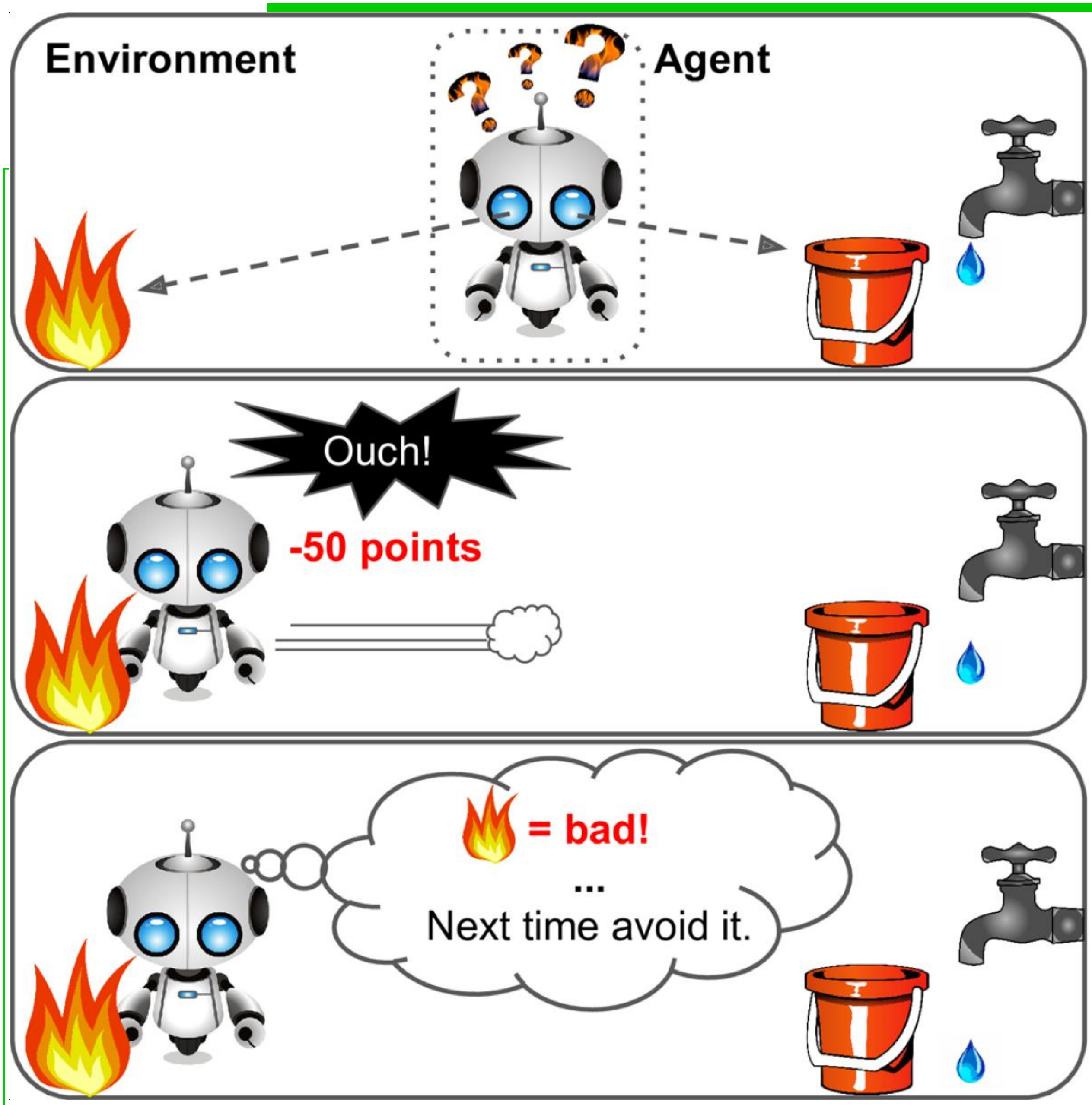
Types of Machine Learning Systems



Types of Machine Learning Systems

- **Semi-supervised learning dealing with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data.**

Example: photo-hosting services , such as Google Photos.
Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in photos 2, 5, and 7.

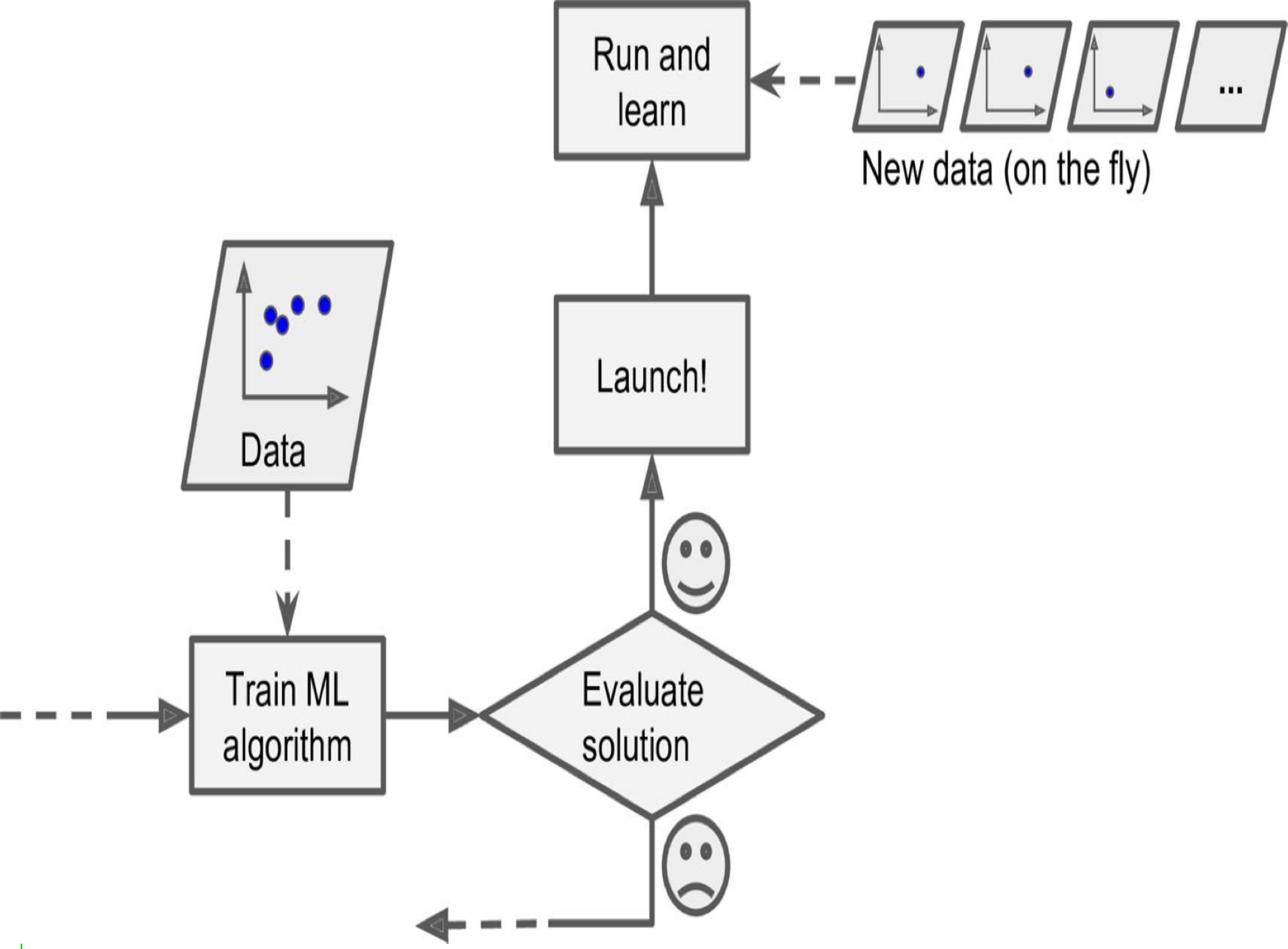


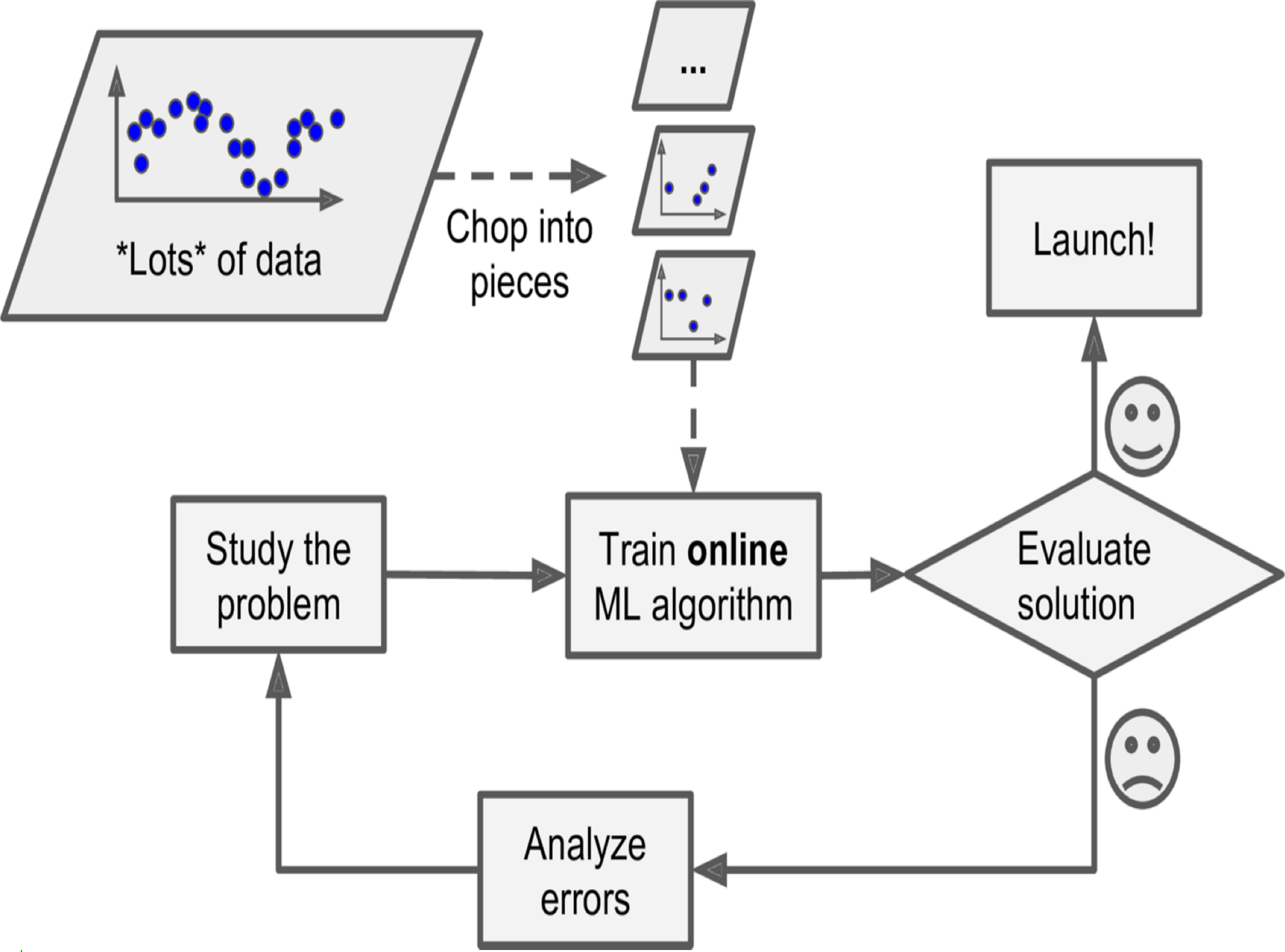
1. Observe
2. Select action using policy
3. Action!
4. Get reward or penalty
5. Update policy (learning step)
6. Iterate until an optimal policy is found

Batch and Stream Learning

- In batch learning, the system is incapable of learning incrementally: it must be trained using all the available data. It take a lot of time and computing. (see next slide)

- In online learning, train the system incrementally by feeding it data instances sequentially, either individually or by small groups called minibatches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives (e.g., stock prices).


Run and learn
...
New data (on the fly)
Data
Launch!
Train ML algorithm
Evaluate solution


Lots of data
Chop into pieces
...
Launch!
Study the problem
Train **online** ML algorithm
Evaluate solution
Analyze errors

Main Challenges of Machine Learning

- **Insufficient Quantity of Training Data**
- **Non-representative Training Data**
- **Poor-Quality Data**
- **Irrelevant Features**
- **Overfitting the Training Data**
- **Underfitting the Training Data**
- **One of important issues for machine learning - how to train a ML model so as to get best performance?**

Testing and Validating

- One way - to put your model in production and monitor how well it performs. but if the model is horribly bad, users will complain—not good idea.

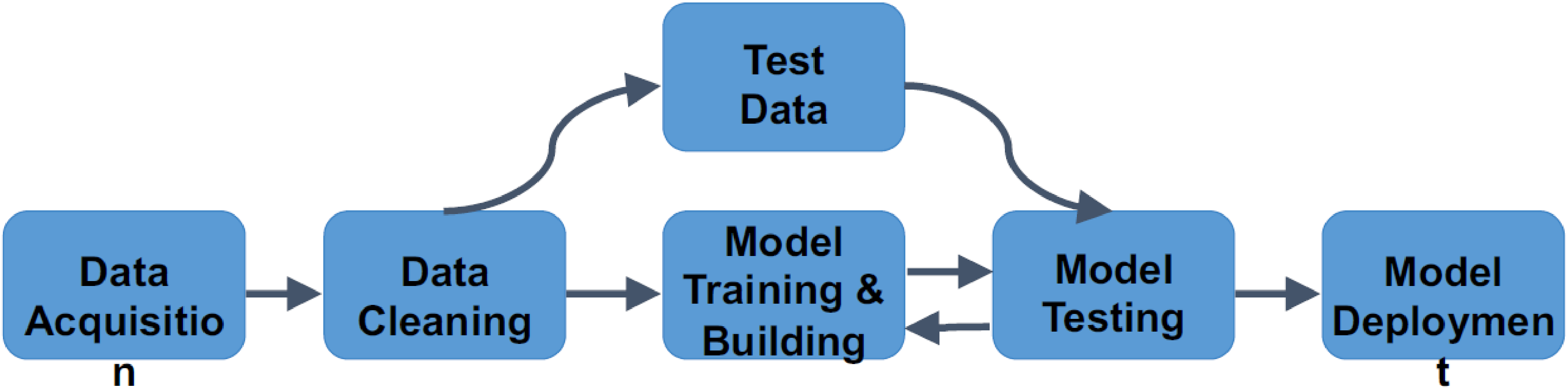
- A better option is to split your data into two sets: the training set (80% of the data for training a model) and the test set (20% of the data for testing the model).

Testing and Validating

- The error rate on new cases is called the generalization error (or out of sample error), and by evaluating your model on the test set, you get an estimation of this error, which tells you how well your model will perform on instances it has never seen before.

- If the training error is low (i.e., your model makes few mistakes on the training set), but the generalization error is high, it means that your model is overfitting the training data.

Machine Learning Process



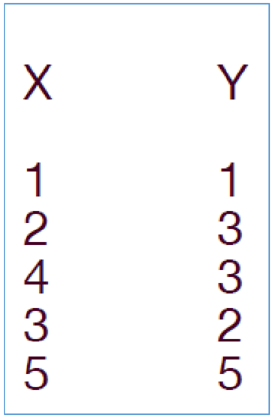
内容提要

Chapter 5: Machine Learning – Overview

Chapter 6: ML Using Linear Regression

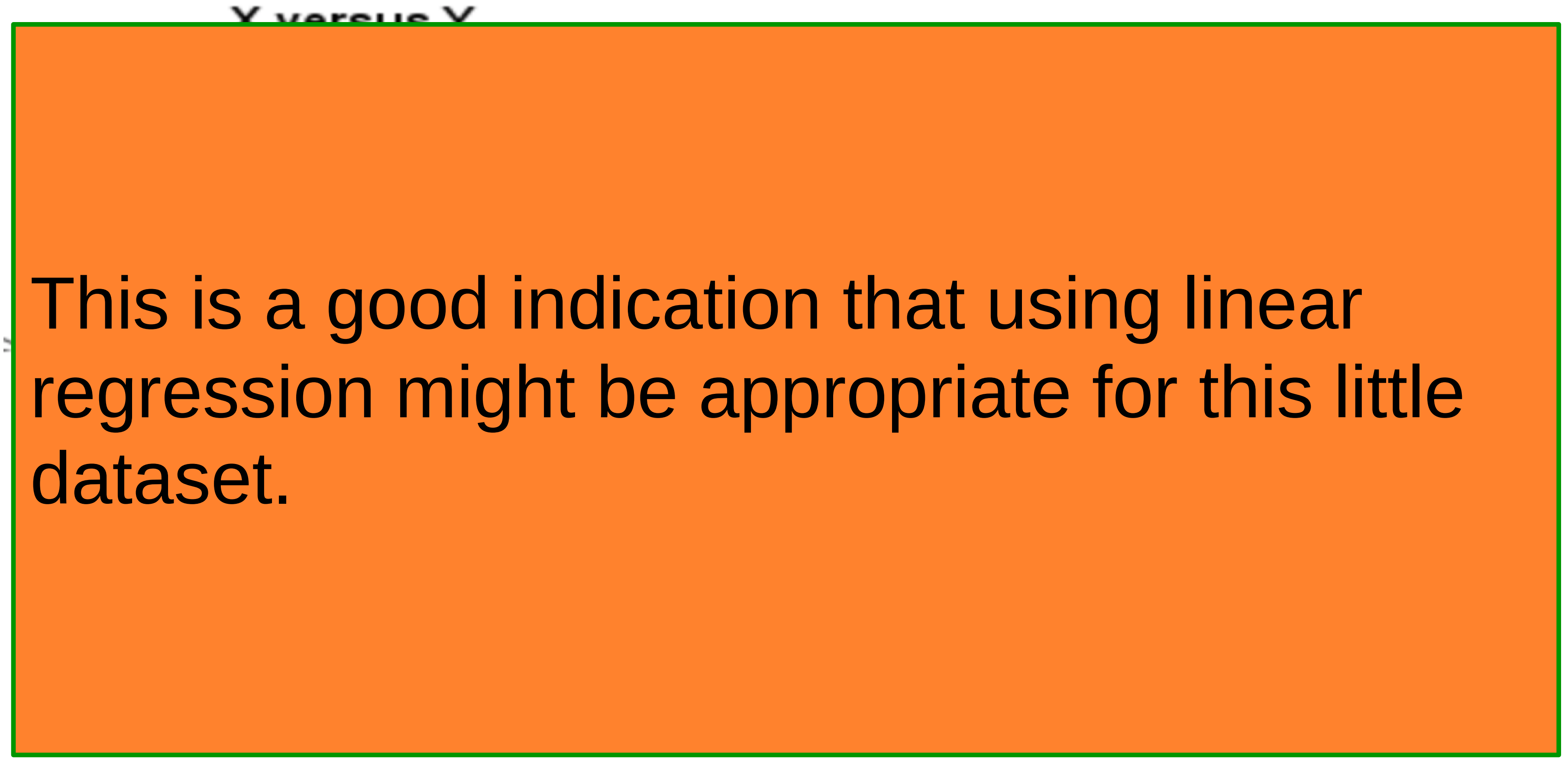
Chapter 6: ML Using Linear Regression

- **Linear regression: a very simple method but widely in use. How it works specifically?**
  - 1) It estimates the coefficients for a simple linear regression model from your training data.
  - 2) It makes predictions using your learned model.
- **Let us see an example.**
- **1. Training data**
  - The attribute X is the input variable and Y is the output variable that we are trying to predict.
  - If we got more data, we would only have X values and we would be interested in predicting Y values.



X	Y
1	1
2	3
4	3
3	2
5	5

ML Using Linear Regression



This is a good indication that using linear regression might be appropriate for this little dataset.

Plot of the Dataset for Simple Linear Regression

Simple Linear Regression

- When we have a single input attribute (x) and we want to use linear regression, this is called simple linear regression

- If we had multiple input attributes (e.g. x1, x2, x3, etc.) This would be called multiple linear regression.

- Here, a simple linear regression is a place to start.

Simple Linear Regression

- **Objective:**

“To create a simple linear regression model from our training data, then make predictions for our training data to get an idea of how well the model learned the relationship in the data.”

- Modeling our data as follows:

$$Y = b0 + b1X$$

- where Y is the output variable we want to predict, X is the input variable we know and b0 and b1 are coefficients , unknown, to be estimated.

Simple Linear Regression

- Technically, b0 is called the intercept because it determines where the line intercepts the y-axis.

- In ML, it is called as bias, because it is added to offset all predictions that we make.

- The b1 is called the slope because it defines how x translates into a y value before we add our bias.

The Goal

- **“To find the best estimates for the coefficients to minimize the errors in predicting Y from X.”**
- **i). Estimating the value for b1 as:**

Where,

$$b_1 = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sum(X - \bar{X})^2}$$

$$\bar{X} = mean(X)$$
$$\bar{Y} = mean(Y)$$

$$b_0 = Y - b_1 X$$

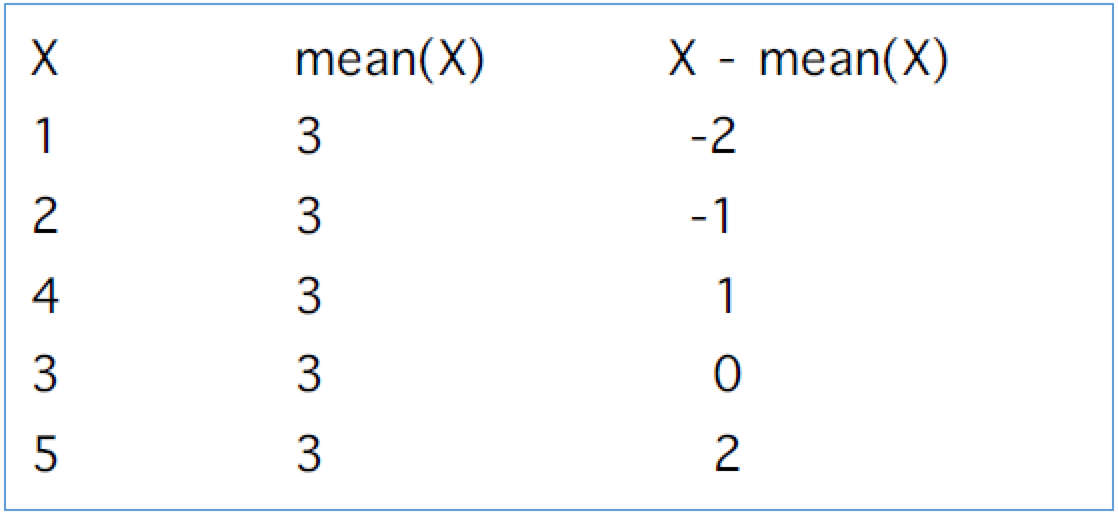
- We can calculate b0 using b1 based on statistics.

The Goal

- **Let’s calculate the mean value of X and Y variables:**

$$\bar{X} = 3 \qquad \bar{Y} = 2.8$$

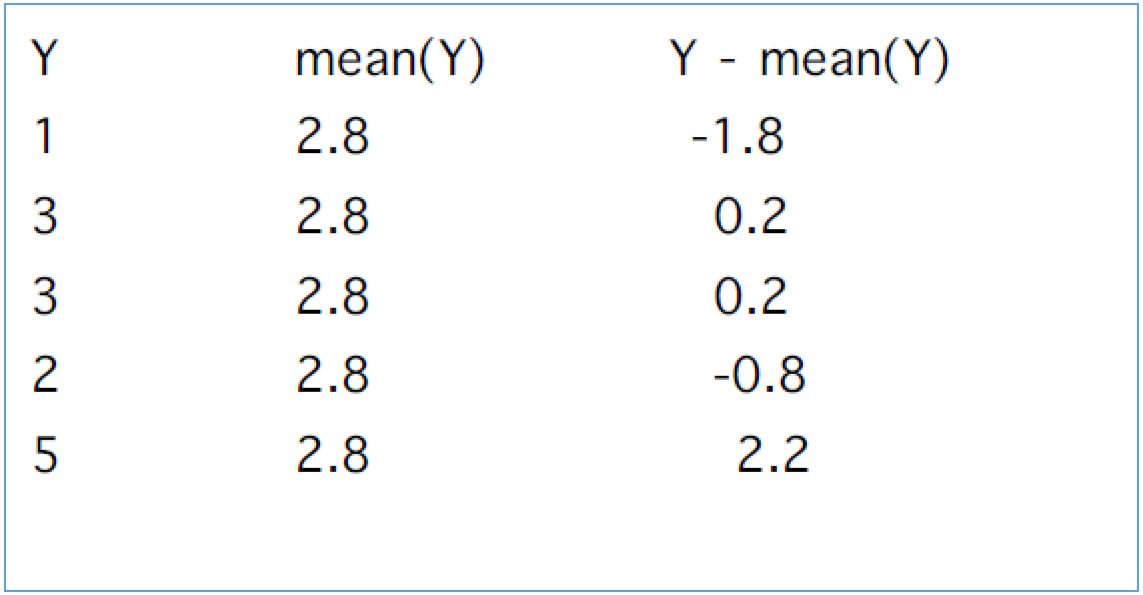
- Now we need to calculate the error of each variable from the mean.
- Error with X:



X	mean(X)	X - mean(X)
1	3	-2
2	3	-1
4	3	1
3	3	0
5	3	2

The Goal

- Error with Y:



Y	mean(Y)	Y - mean(Y)
1	2.8	-1.8
3	2.8	0.2
3	2.8	0.2
2	2.8	-0.8
5	2.8	2.2

- So, we are able to calculate b1 now, given mean values and error, as above..

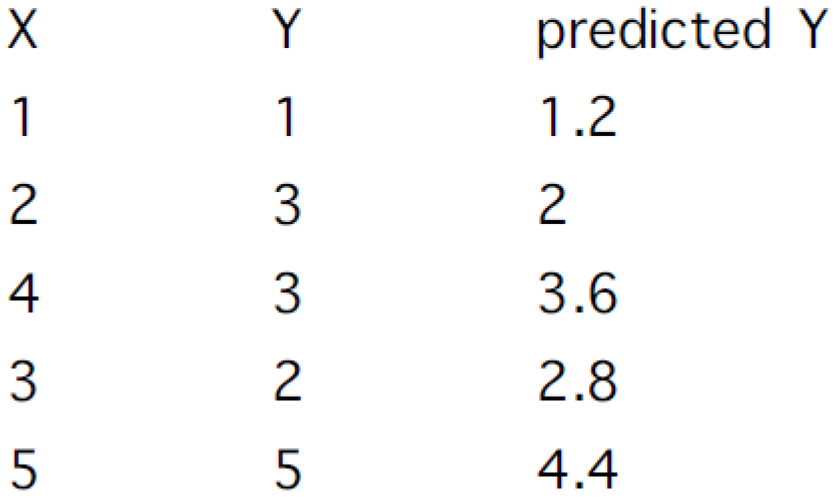
The Goal

- ii). Estimating The Intercept b0:
  - Easier, as already know the values of all of the terms involved.

$$\boldsymbol{b_0 = Y - b_1 X}$$

- **iii) Making Predictions**
- We now have the coefficients for our simple linear regression equation.

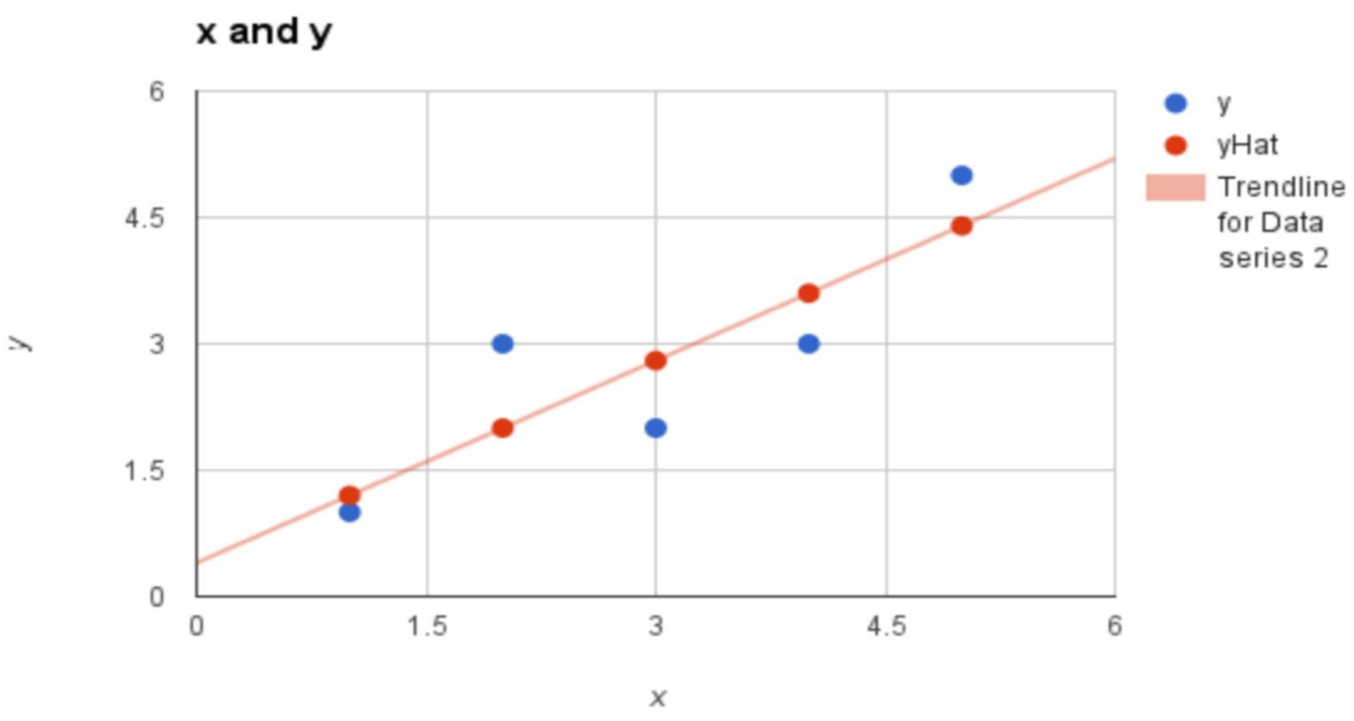
$$\boldsymbol{Y = b_0 + b_1 X}$$



X	Y	predicted Y
1	1	1.2
2	3	2
4	3	3.6
3	2	2.8
5	5	4.4

The Goal

- Plot these predictions as a line with our data. This gives us a visual idea of how well the line models our data.



Simple Linear Regression Model

The Goal

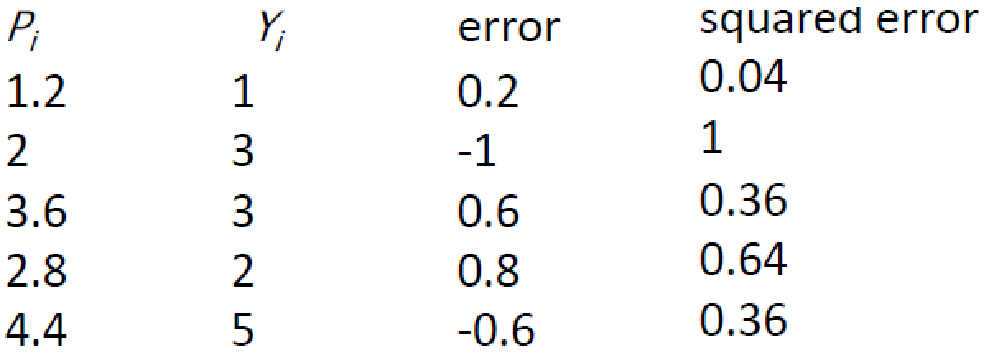
- **iv. Estimating Error**
  - We can calculate a error for our predictions called the Root Mean Squared Error(RMSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - Y_i)^2}{n}}$$

  - Where Pi is the predicted value and Yi is the actual value,
  - i is the index for a specific instance,
  - n is the number of predictions, because we must calculate the error across all predicted values.
  - So, RMSE = 0.692,

after evaluated by

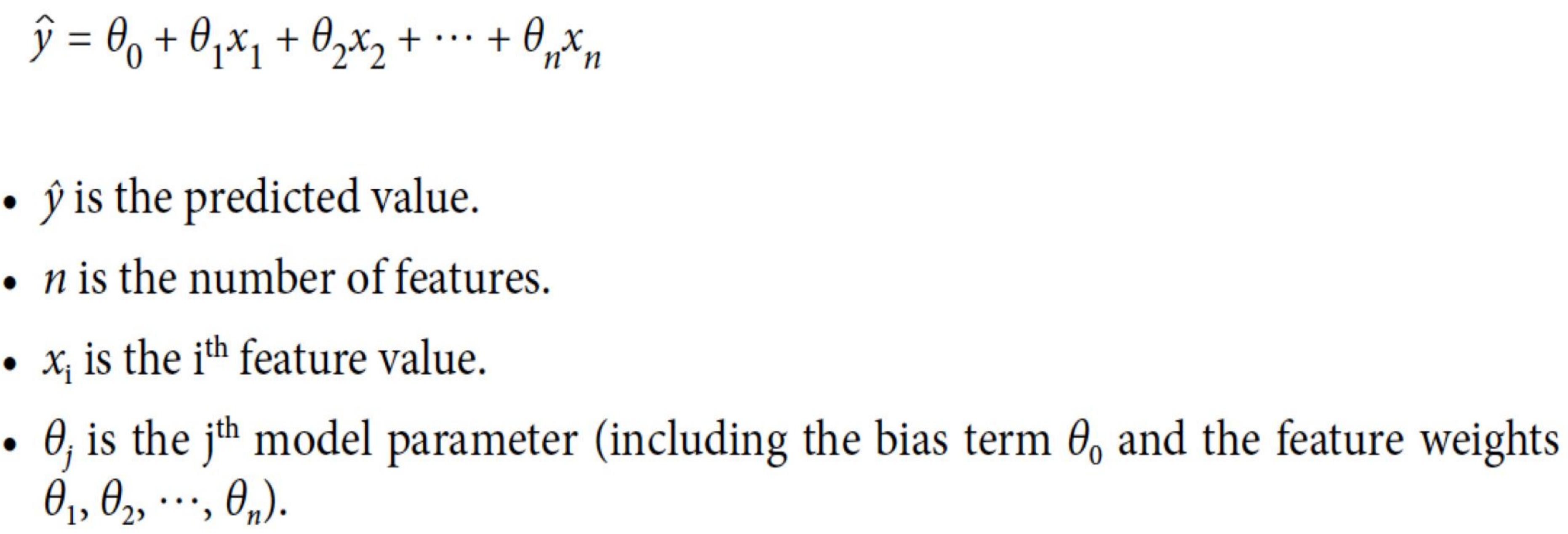
using the equation as above.



P_i	Y_i	error	squared error
1.2	1	0.2	0.04
2	3	-1	1
3.6	3	0.6	0.36
2.8	2	0.8	0.64
4.4	5	-0.6	0.36

The Goal

- **V. General equations in math**
- Linear Regression model prediction

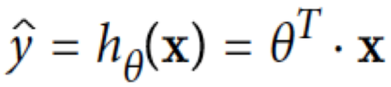


$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

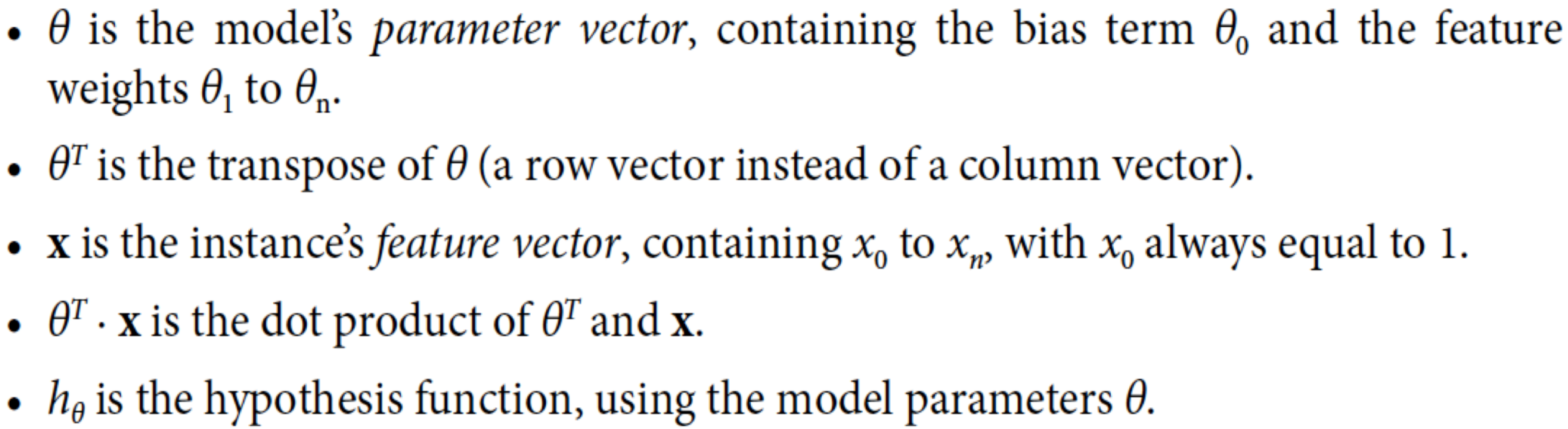
- $\hat{y}$ is the predicted value.
- n is the number of features.
- x_{i} is the ith feature value.
- θ_j is the jth model parameter (including the bias term θ_0 and the feature weights $\theta_1, \theta_2, \cdots, \theta_n$).

The Goal

- **Linear Regression model prediction (vectorized form)**



$$\hat{y} = h_\theta(\mathbf{x}) = \theta^T \cdot \mathbf{x}$$



- θ is the model's *parameter vector*, containing the bias term θ_0 and the feature weights θ_1 to θ_n.
- θ^T is the transpose of θ (a row vector instead of a column vector).
- $\mathbf{x}$ is the instance's *feature vector*, containing x_0 to x_n, with x_0 always equal to 1.
- $\theta^T \cdot \mathbf{x}$ is the dot product of θ^T and $\mathbf{x}$.
- h_θ is the hypothesis function, using the model parameters θ.

The Goal

- To find the value of θ that minimizes the cost function, there is a closed-form solution, in other words, a mathematical equation that gives the result directly. This is called the Normal Equation:

$$\hat{\theta} = \left(\mathbf{X}^T \cdot \mathbf{X}\right)^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$$

- $\hat{\theta}$ is the value of θ that minimizes the cost function.
- $\mathbf{y}$ is the vector of target values containing $y^{(1)}$ to $y^{(m)}$.

The Goal

- **MSE cost function for a Linear Regression model**

$$\mathrm{MSE}\left(\mathbf{X}, h_{\theta}\right) = \frac{1}{m} \sum_{i=1}^{m} \left(\theta^{T} \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^{2}$$

3. Machine Learning Library (MLlib)

- MLlib is Spark’s machine learning (ML) library.

- Its goal is to make practical machine learning scalable and easy. It consists of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as lower-level optimization primitives and higher-level pipeline APIs.

3. Machine Learning Library (MLlib)

- **It divides into two packages:**
  - spark.mllib contains the original API built on top of RDDs.
  - spark.ml provides higher-level API built on top of DataFrames for constructing ML pipelines, which will be used in this course.

4. Scikit-Learn API for Machine Learning

- Scikit-learn is a free software machine learning library for the Python programming language and data analysis.
- It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k -means and DBSCAN.
- It is designed to interoperate with Pandas, NumPy, SciPy, and matplotlib.
- It is open source, commercially usable - BSD license.

结束

2020年9月22日