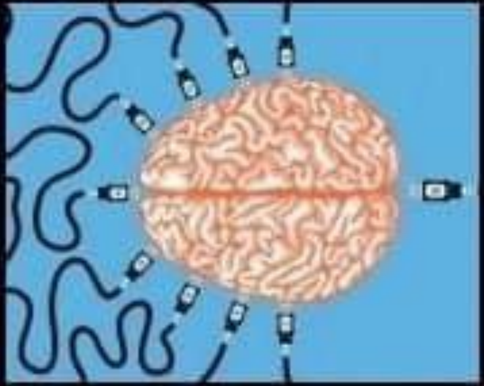
Machine Intelligence

Introduction to Machine Learning
Geron Chapter 1

Machine Learning

What society thinks I do.



What my friends thinks I do.

What computer scientists think I do.

What my boss thinks I do.

What I think I do.

What I really do.

Machine Learning has been around for awhile!

- Wikipedia to the rescue: https://en.wikipedia.org/wiki/Machine_learning

- ***Definitions***:
  - Machine Learning is the science (and art) of programming computers so they can learn from data
  - ***Slightly more general***: Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. (Arthur Samuel, 1959)
  - ***Engineering oriented***: A computer program is said to learn from experience E, with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. (Tom Mitchell, 1997)

Spam filter example of machine learning

- A spam filter is a machine learning program that can learn to flag spam given examples of spam emails (e.g., flagged by users) and examples of regular emails.

- ***Training Set***: The examples the system uses to learn
  - ***Training Instance (sample)***: Each training example the system uses to learn

- The task T is to flag spam for new emails.
  - The experience E is the training data

- ***Performance measure P*** needs to be defined
  - Example: The ratio of correctly classified emails – this performance measure is called accuracy and is often used in classification tasks

Why use machine learning?

- Consider how to write a spam filter with traditional programming techniques
  - ***First***: look at what spam typically looks like. We notice certain words tend to show up (4U, credit card, free, amazing). Perhaps other patterns as well
  - ***Second***: Write a detection algorithm for each of the patterns you noticed. Your program flags emails as spam if some number of these patterns are detected
  - ***Third***: Test the program repeating steps 1 and 2 until it is good enough

- Since the problem is complex, our program is going to have a long list of rules – which will be difficult to maintain

Why use machine learning?

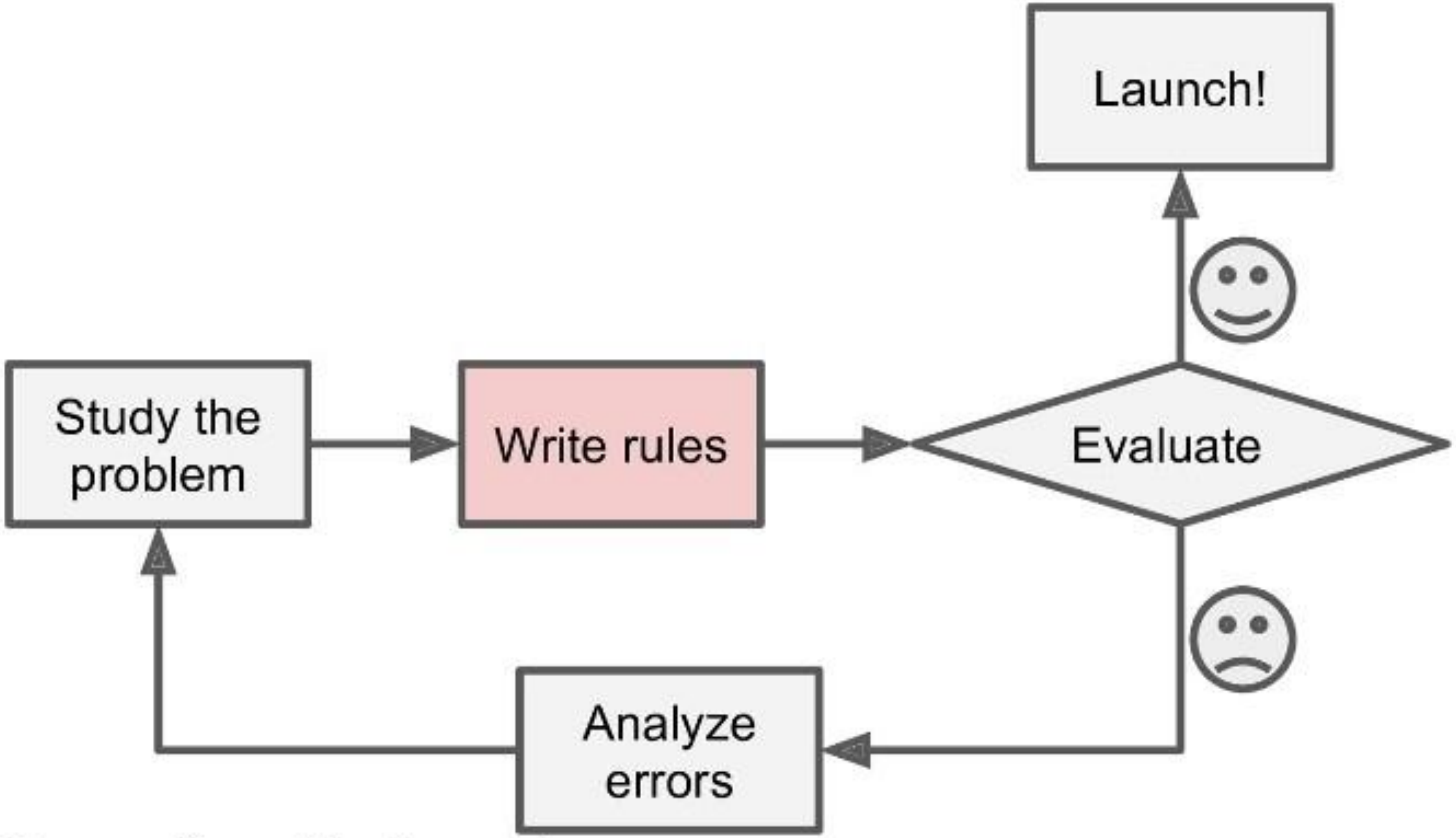


Figure 1-1. The traditional approach

Why use machine learning?

- Compare to a machine learning program
  - The program ***automatically learns which words and phrases are good predictors of spam*** by detecting unusual patterns of words in the spam examples
  - The program is much shorter and easier to maintain – and likely more accurate

- And what if the spammers realize what is going on and start using different words?

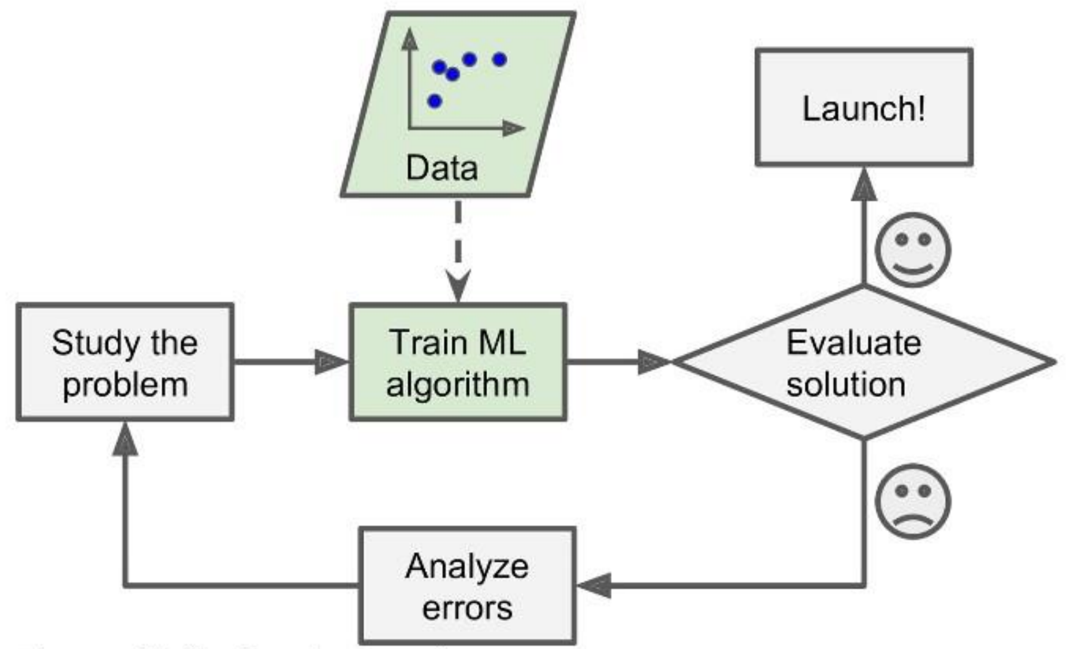


Figure 1-2. Machine Learning approach

Why use machine learning?

- ML programs can automatically adjust to changes in word frequency in spam email

- *Example*: Statistical machine translation between languages
  - https://en.wikipedia.org/wiki/Statistical_machine_translation

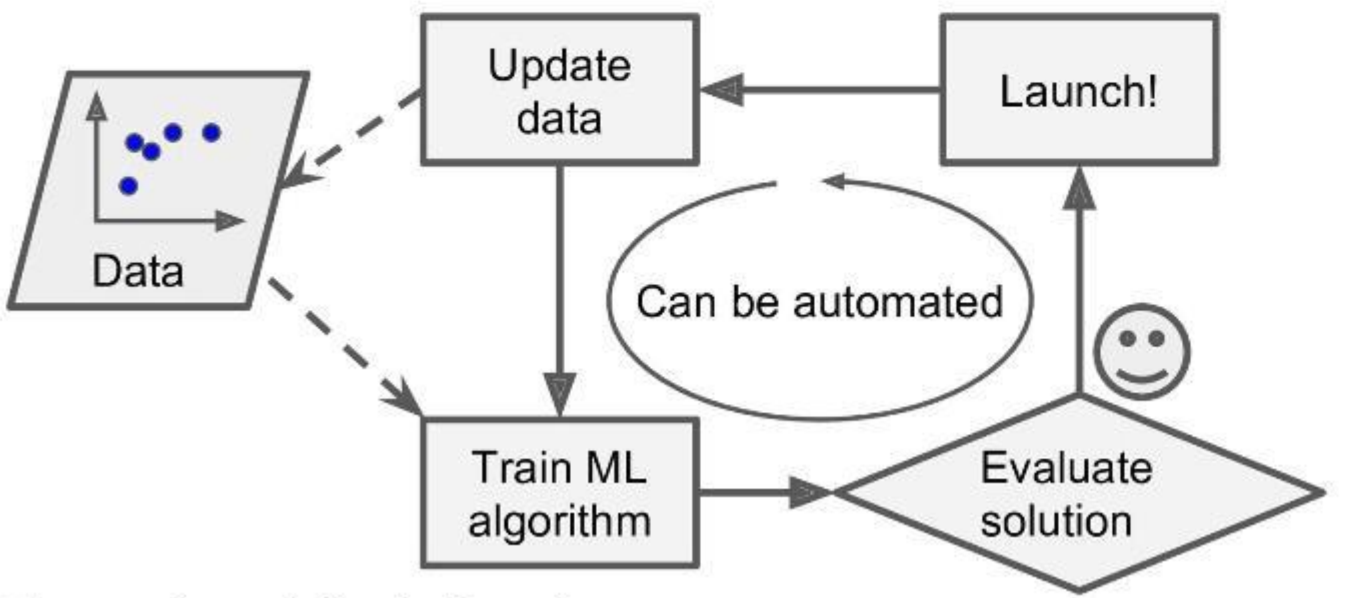


Figure 1-3. Automatically adapting to change

What is *data mining*?

- Applying ML techniques to dig into large amounts of data can help ***discover patterns that were not immediately apparent***

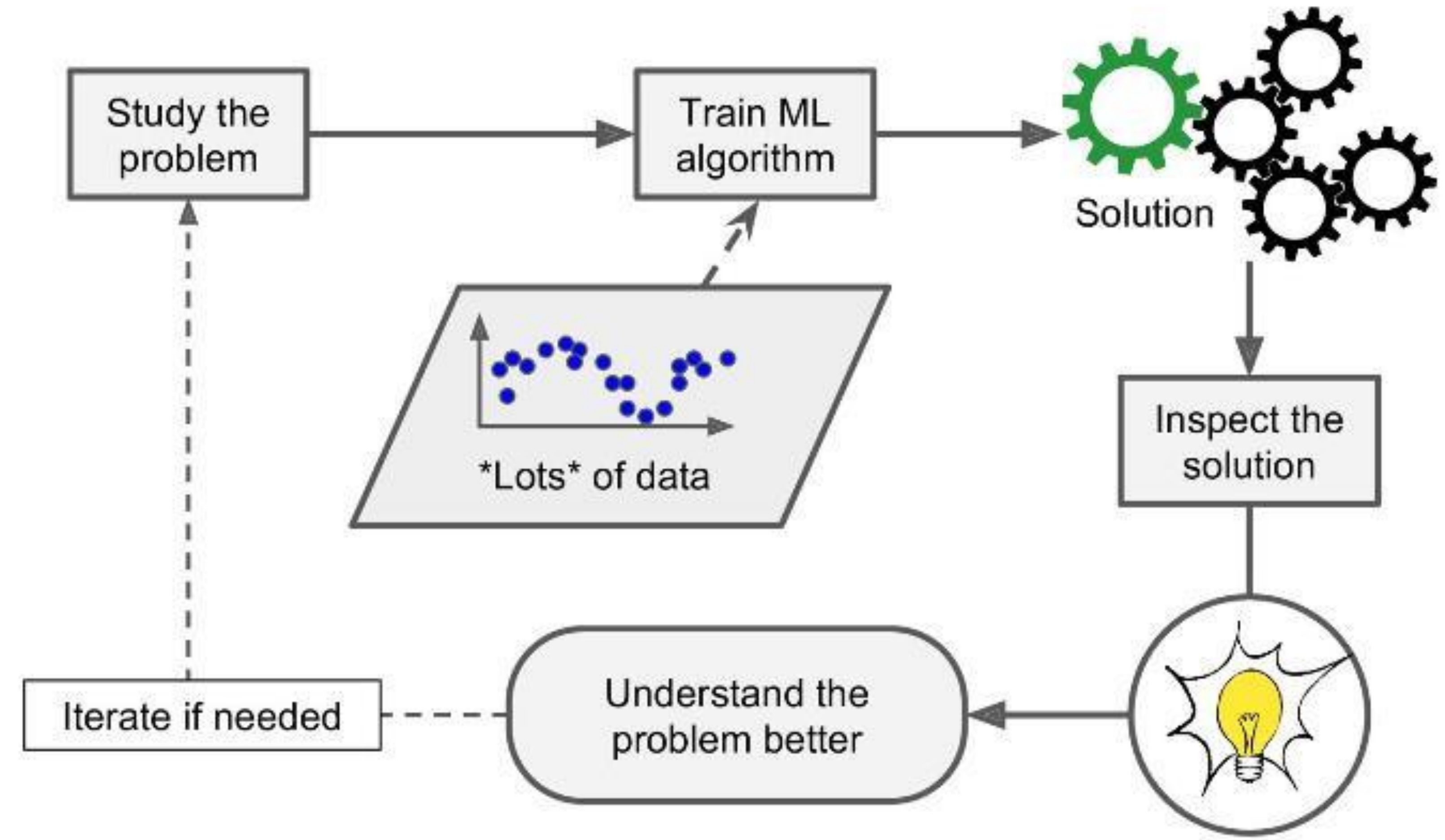


Figure 1-4. Machine Learning can help humans learn

Types of Machine Learning Systems

- Trained with human supervision?
- Can learn incrementally on the fly (online vs batch learning)
- Do they simply compare new data points to known data points, or instead detect patterns in the training data and build a predictive model
- These criteria are not exclusive. You can combine them any way you like

Supervised learning

- ***Supervised Learning***: The training data includes the desired solutions – called labels
  - A typical supervised learning task is classification (spam filter a good example)
  - Another typical task is to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.) called predictors. This is called a regression task
- ***Attribute***: is a data type, e.g., mileage
- ***Feature***: generally, means an attribute plus its value – mileage = 15,000

Unsupervised learning

- The training data is unlabeled – no answers provided in the training set
- Types of unsupervised learning:
  - ***Clustering***: Detect groups of similar features
  - ***Visualization and dimensionality reduction***: Feed them lots of complex and unlabeled data. They output a 2D or 3D representation of the data that can be plotted
  - ***Association rule learning***: Dig into large amounts of data and discover interesting relations between attributes.

Semi-supervised learning

- Some algorithms can deal with ***partially labeled training data***, usually a lot of unlabeled data and a bit of labeled data

- Google Photos is a good example of this.
  - It automatically recognizes the same person in different photos
  - Once you tell Google Photos who a person is, it labels that person in all the photos they are in

Reinforcement learning

- The ***learning system is called an agent***. It observes the environment, selects and performs actions, and gets rewards in return (or penalties as negative rewards. It ***must learn by itself what is the best strategy***, called a policy, to get the most reward over time.

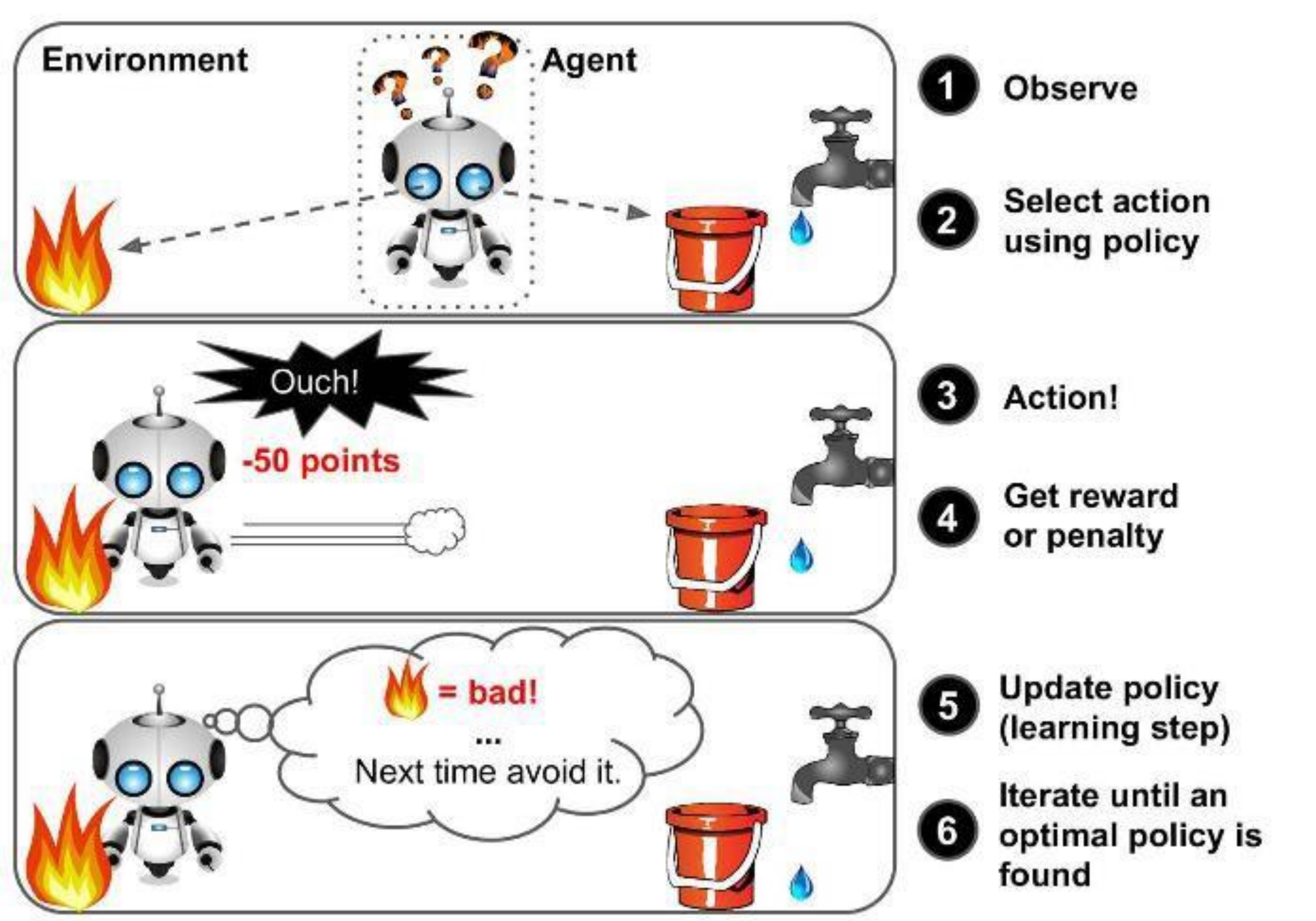


Figure 1-12. Reinforcement Learning

Batch and Online learning

- ***Batch learning***: The system is unable to learn incrementally
  - The system must learn everything first, then it runs in production without more learning – also called ***offline learning***
  - To have the system “learn more” you must train a new version of the system

- ***Online learning***: The system is trained incrementally by feeding it data instances sequentially.
  - Each learning step is fast and cheap, so the system can learn about new data as it arrives
  - ***Good example***: Stock price predictors – continuous flow of data
  - ***Learning rate***: how fast should the system adapt to new data?

Online learning diagram

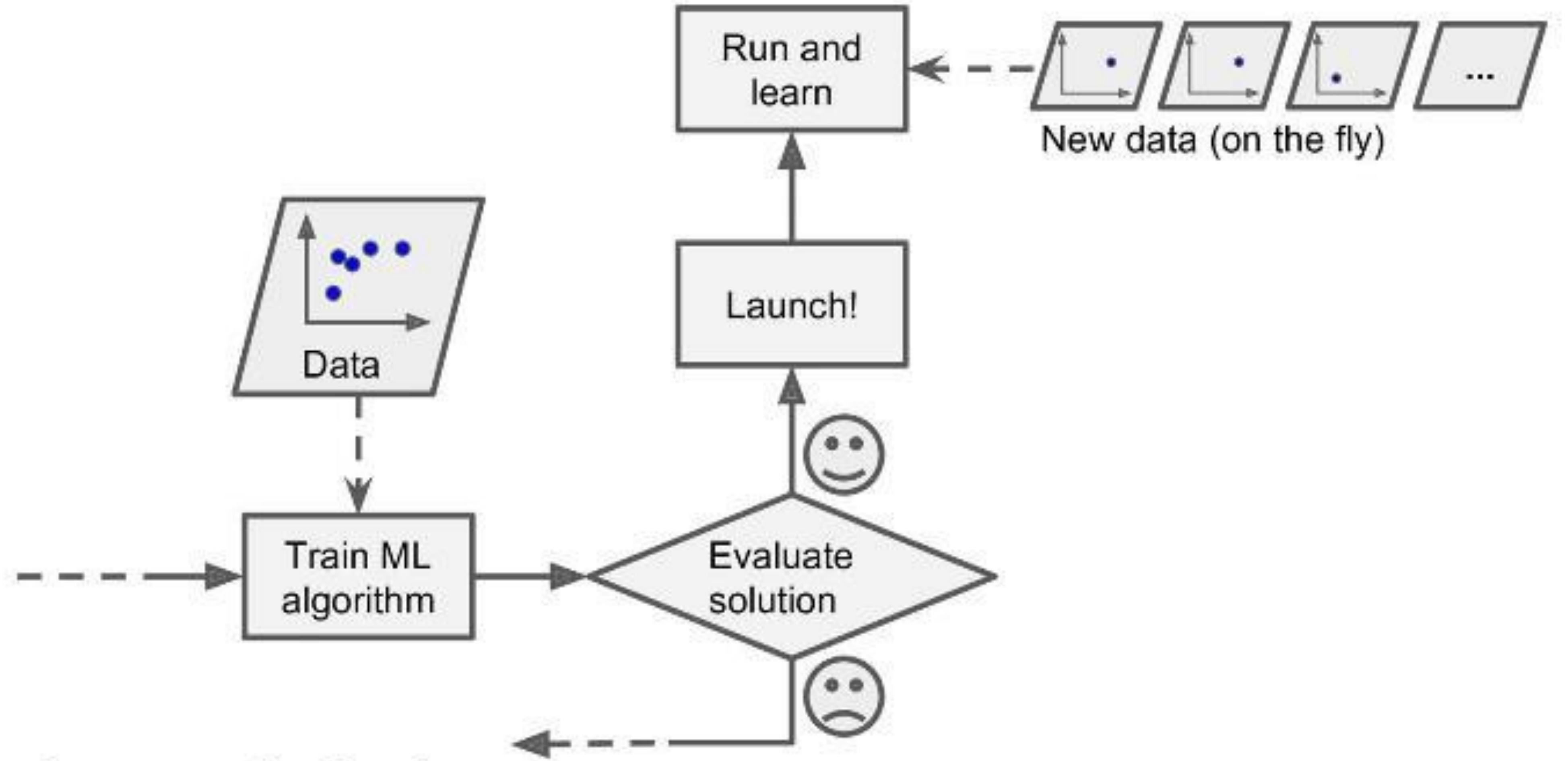


Figure 1-13. Online learning

Instance-Based versus Model-Based Learning

- Another way to categorize machine learning systems is by how they ***generalize***

- Most machine learning tasks are about making predictions
  - This means that given several training examples, the system needs to be able to generalize to examples it has never seen before
  - Having a good performance measure on the training data is good, but insufficient
  - The true goal is to perform well on new instances
- ***Instance-Based Learning***: System learns the examples by heart, then generalizes to new cases using a similarity measure

Instance-Based Learning

- System learns the examples by heart, then generalizes to new cases using a similarity measure
  - Instead of just flagging emails identical to known spam emails, our spam filter would be programmed to also flag emails that are like known spam emails.
  - For instance, count the number of words in common

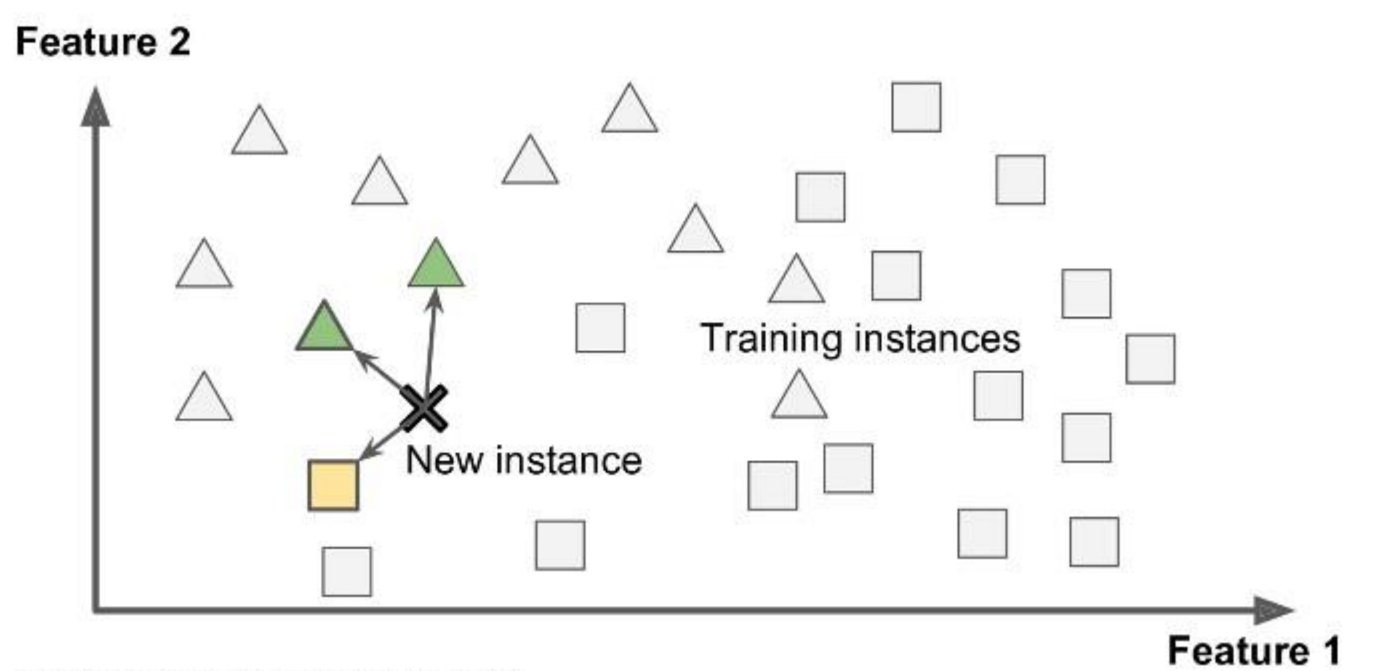


Figure 1-15. Instance-based learning

Model-Based Learning

- Another way is to build a model of these examples, then use that model to make predictions.

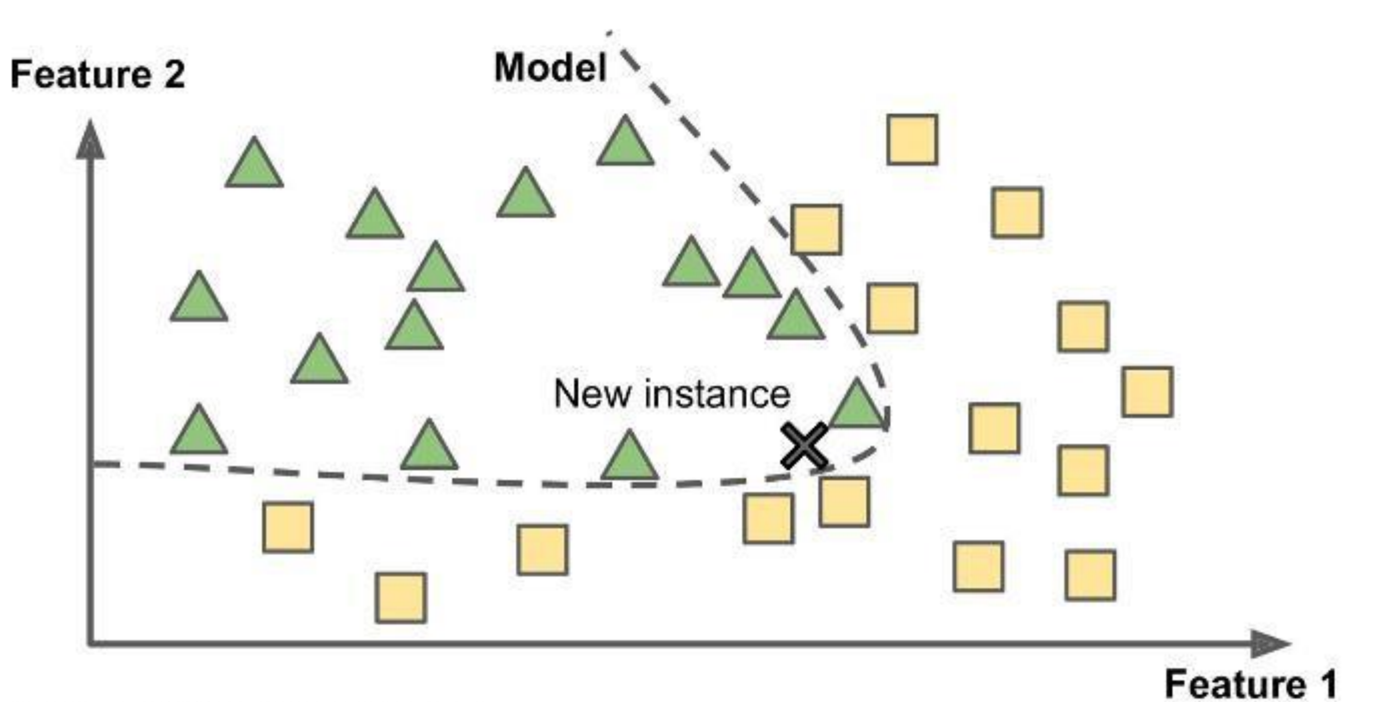


Figure 1-16. Model-based learning

Model-Based Learning example

- Suppose you want to know if money makes people happy
  - Let's use a sample of the Better Life Index data from OECD's website (https://www.oecd.org/, Organization for Economic Cooperation and Development) and statistics about GDP per capita from IMF's website.
- Do you see a pattern?

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

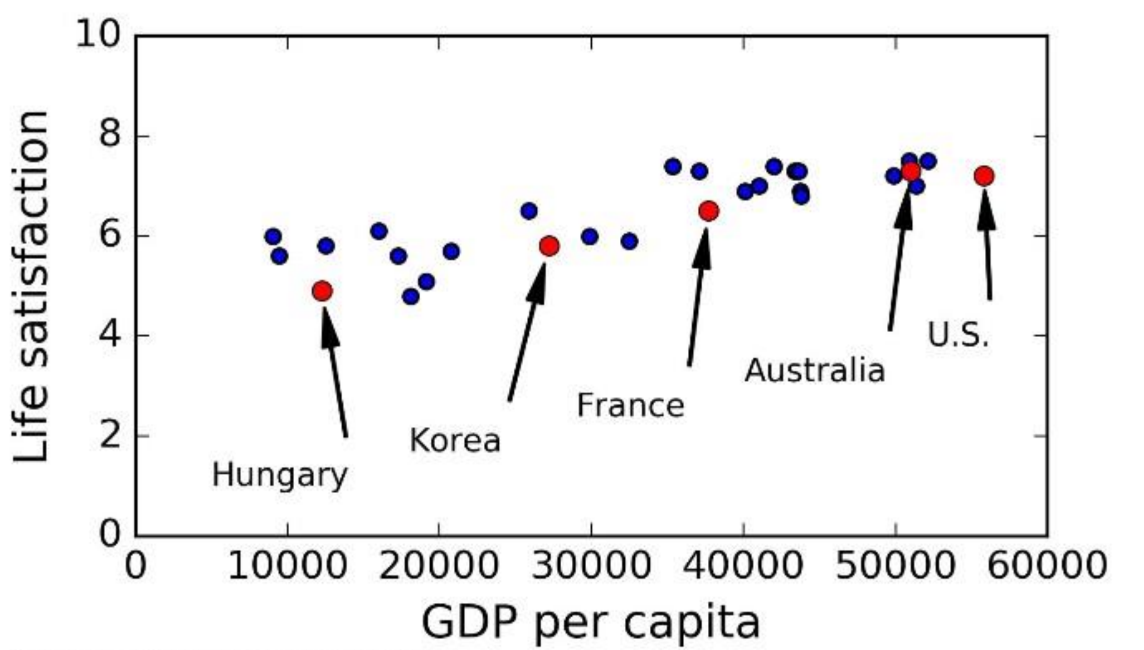


Figure 1-17. Do you see a trend here?

Mode-Based Learning example continued

- It seems like the data increases linearly with GDP per capita

- This is called ***model selection*** – we are choosing a linear model of our data

Equation 1-1. A simple linear model

$$\text{life\_satisfaction} = \theta_0 + \theta_1 \times \text{GDP\_per\_capita}$$

- Our model has two modal parameters, Θ_0 and Θ_1

- By tweaking those two parameters, we can make our model represent any linear function

Possible linear models

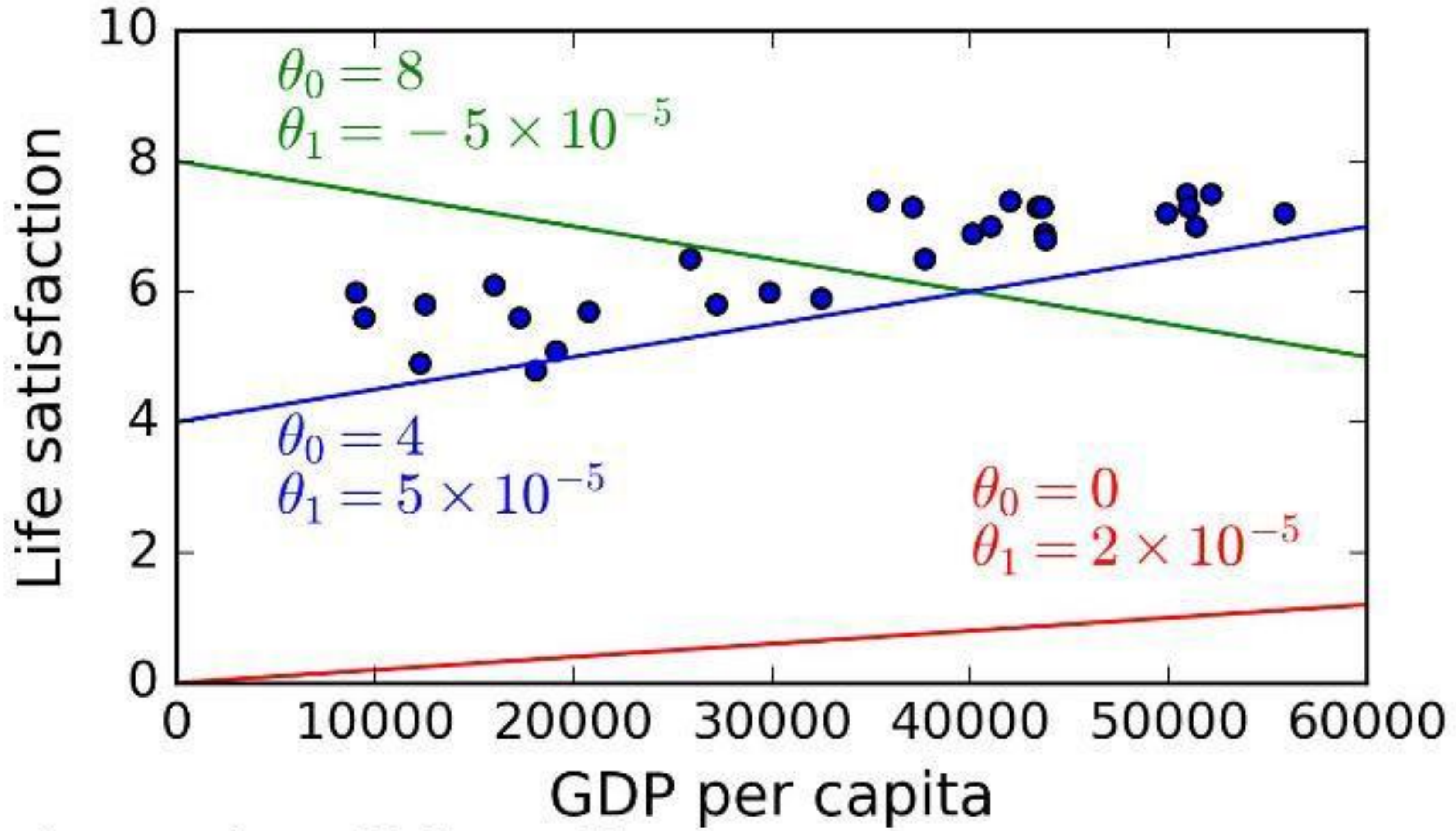


Figure 1-18. A few possible linear models

Model-Based Learning example continued

- But what is the best model? We need to specify a ***performance measure***
  - We can define a ***utility function*** (or fitness function) that measures how good your model is
  - Or we can define a ***cost function*** that measures how bad it is

- For ***linear regression problems***, people typically use a cost function that measures the distance between the linear model's predictions and the training example
  - ***Objective***: Minimize this distance
  - Linear regression models to save the day

Best linear model

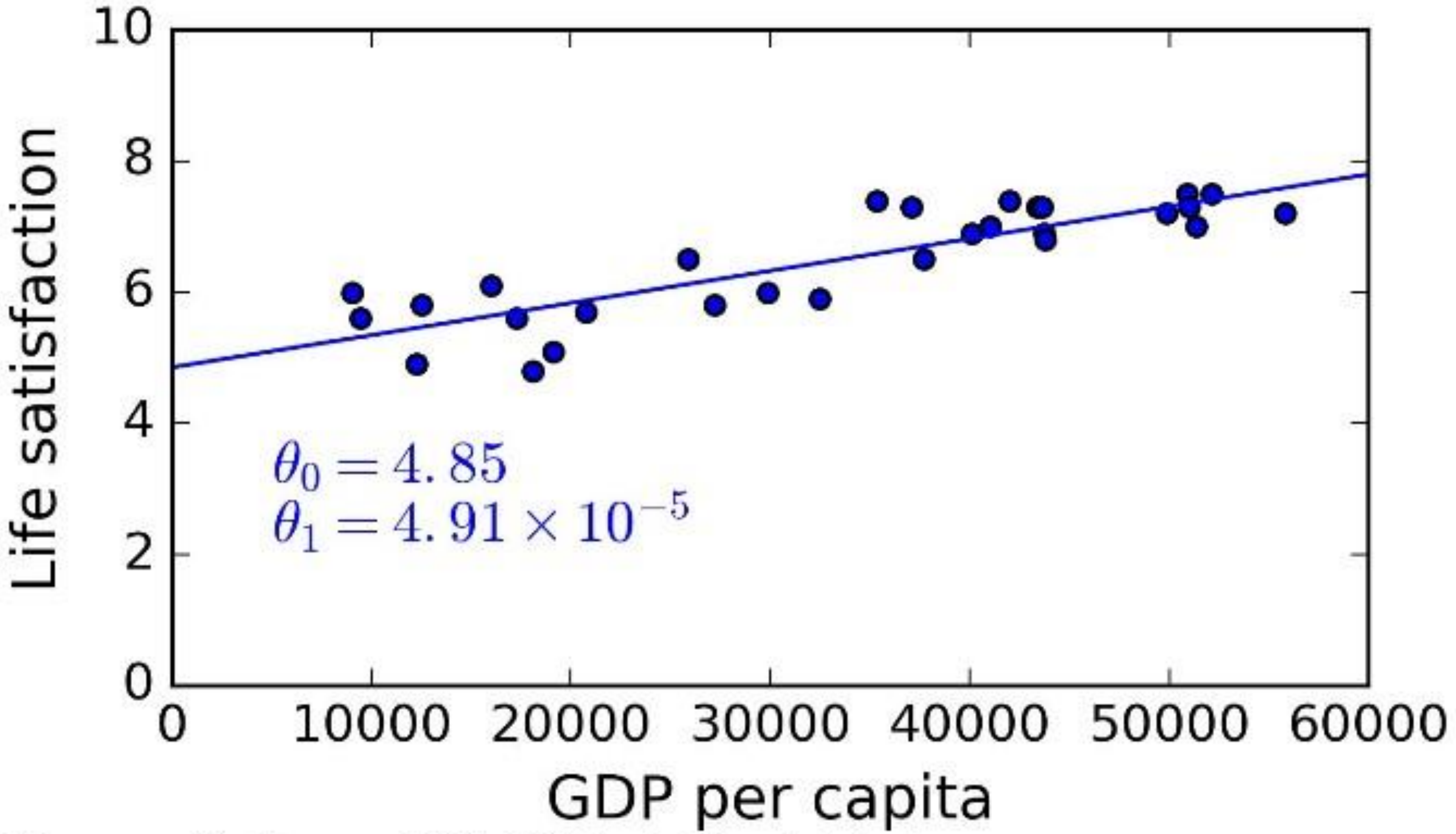


Figure 1-19. The linear model that fits the training data best

Main challenges of machine learning

- ***Two main problems***: bad algorithm and/or bad data
- ***Insufficient Quantity of Training Data***: We may need millions of training instances
- ***Non-representative Training Data***: We train a model that is unlikely to make accurate predictions
- GDP per capita with more data
  - Dashed line is old model
  - Solid line is new model

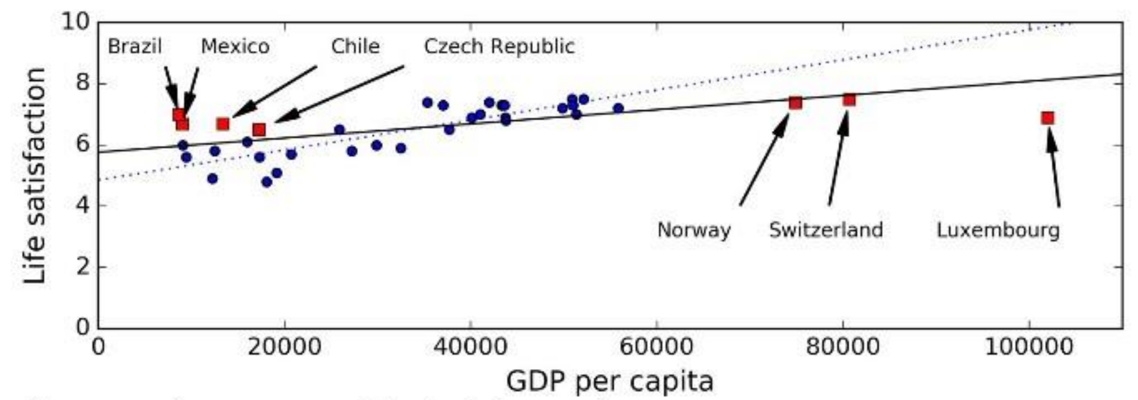


Figure 1-21. A more representative training sample

Examples of bad data

- ***Poor Quality Data***: We may need to clean up the data we have
  - Discard clear outliers
  - Some training instances are missing some features – like age is missing
- ***Irrelevant Features***: Garbage in garbage out. This involves feature engineering
  - ***Feature selection***: selecting the most useful features to train on among existing features
  - ***Feature extraction***: Combining features to produce a more useful one (dimensionality reduction)
  - Creating new features by gathering new data

Examples of bad algorithms

- ***Overfitting the training data***: This means the model performs well on the training data, but does not generalize well

- The figure below is a high-degree polynomial life satisfaction model that strongly overfits the training data
  - Would you trust its predictions?

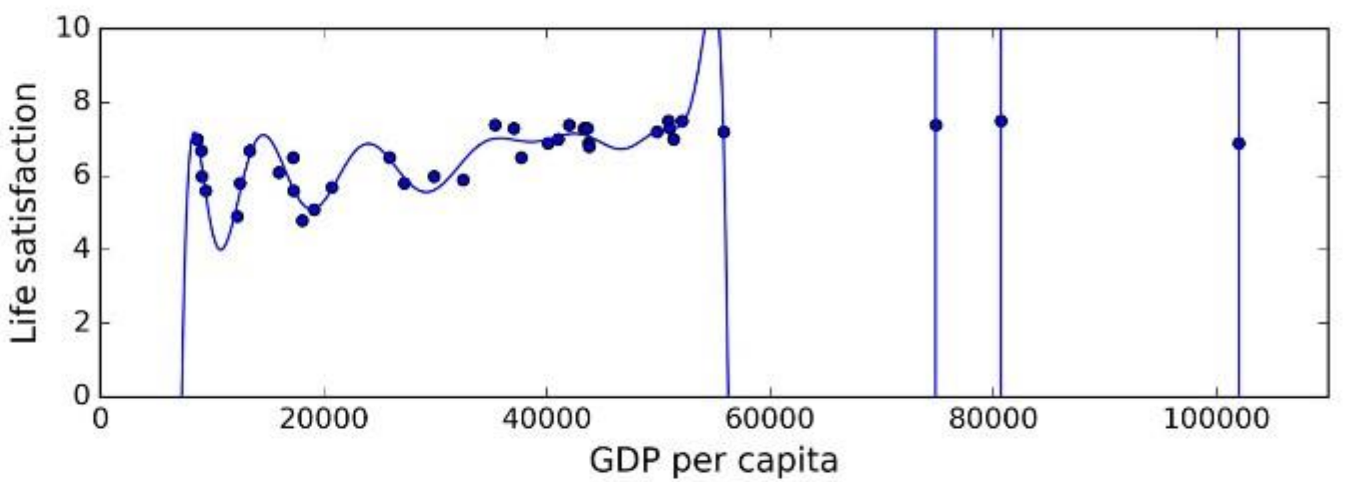


Figure 1-22. Overfitting the training data

Examples of bad algorithms

- ***Underfitting the training data***: Your model is too simple to learn the underlying structure of the data
  - A linear model of life satisfaction is prone to underfit. Reality is just more complex than the model.
  - ***How to solve this problem***:
    - Select a more powerful model, with more parameters
    - Feeding better features to the learning algorithm (feature engineering)
    - Reducing the constraints on the model

Stepping Back

- Machine learning is about ***making machines get better at some task by learning from data***, instead of having to explicitly code rules
- There are ***many different types of machine learning systems***: supervised or not, batch or online, instance-based or model-based, etc.
- In a ML project you ***gather data in a training set, and you feed the training set to a learning algorithm***
- The system will not perform well if your ***training set is too small***, or if the ***data is not representative, noisy, or polluted with irrelevant features***
- Lastly, your ***model needs to be neither too simple nor too complex***

Testing and Validating

- The only way to know how well a model will generalize to new cases is to try it out on new cases
- We can put our model into production and see how it does
- ***Better idea***: Split your training data into two sets – the training set and the test set

Now to start learning the Scikit-Learn Library