CST 407 ML – Lecture 1

Ch 1, What is Machine Learning, hyper-practical approach, regression techniques

# Machine Learning

What is “Machine Learning”?

“Machine Learning is the science (and art) of programming computers so they can learn from data.”

-- Aurélien Géron1

“**Intelligence** can be defined as the ability to acquire and apply knowledge and skills.

**Artificial intelligence** can be defined as the ability of a machine to think like a human being, in order to perform a particular task, without being explicitly programmed.

**Machine learning** can be defined as the process of teaching a machine to think like a human being in order to perform a particular task, without being explicitly programmed.

The entire process of collecting, manipulating, analyzing, and developing inferences from data is known as **data science.”**

-- Nikita Silaparasetty2

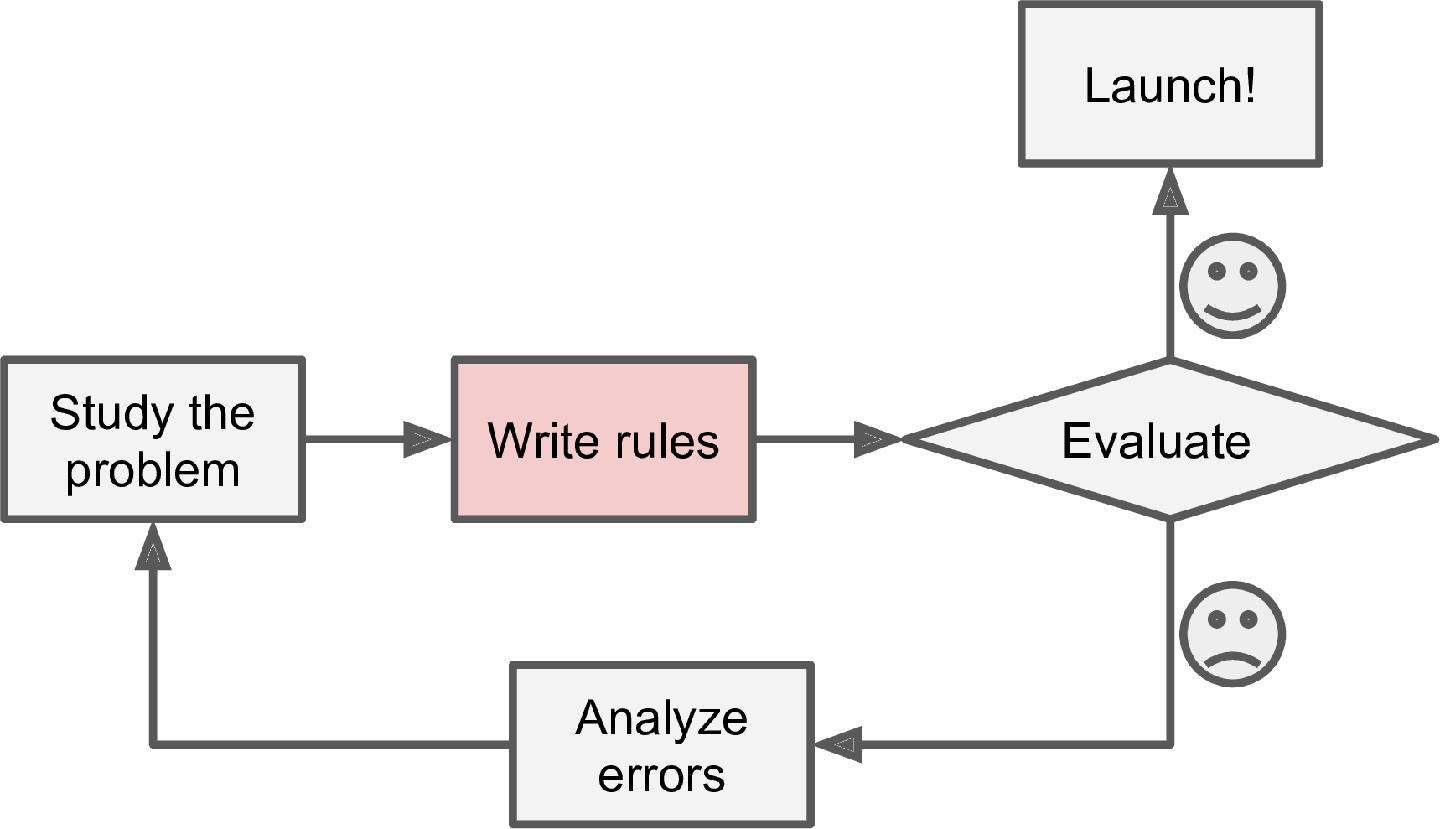
Why do this?

“Today we have many problems for which we lack algorithms but have data.”

-- Ethem Alpaydin3

Example: Spam Filter

If we did this with a traditional algorithm…

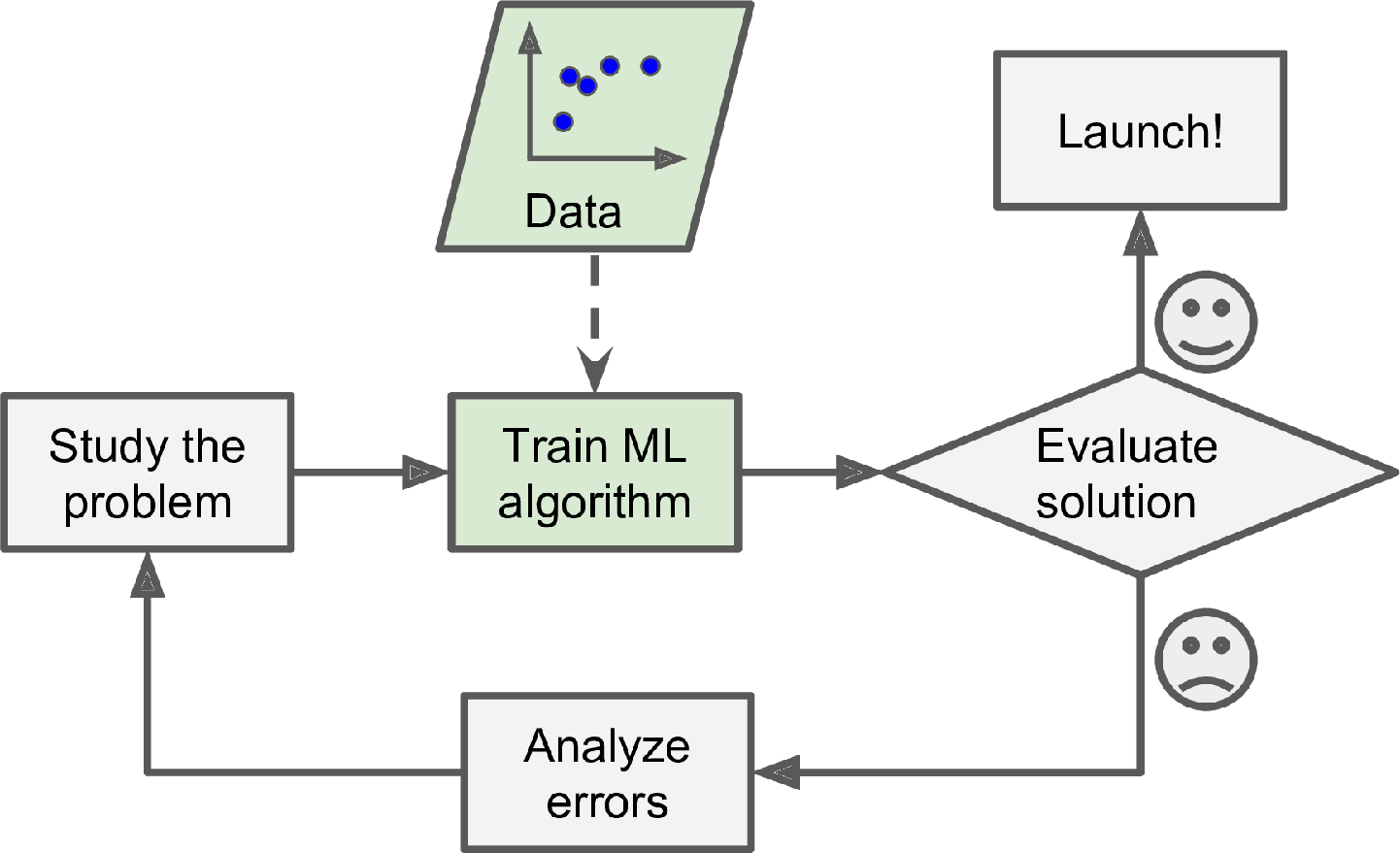


What are the challenges here?

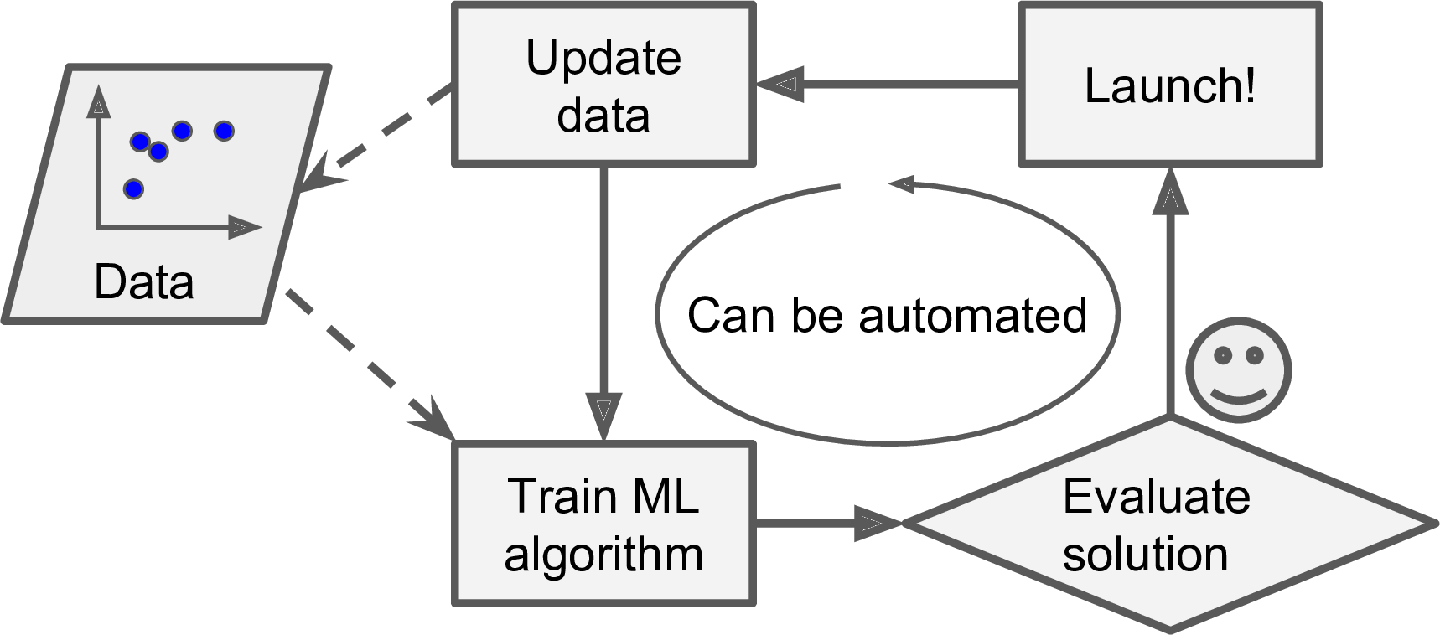
How do we maintain the rules?

How would we solve this problem with Machine Learning?

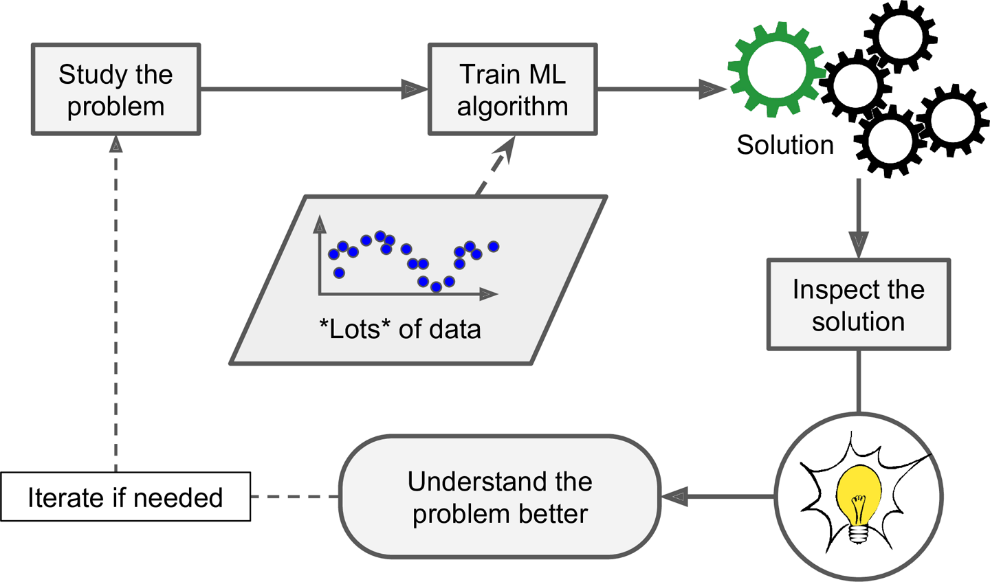
With ML, unmaintainable, hand-coded rules are replaced by rules *learned* from real data…



As Spam changes, the learned rules continue to evolve…



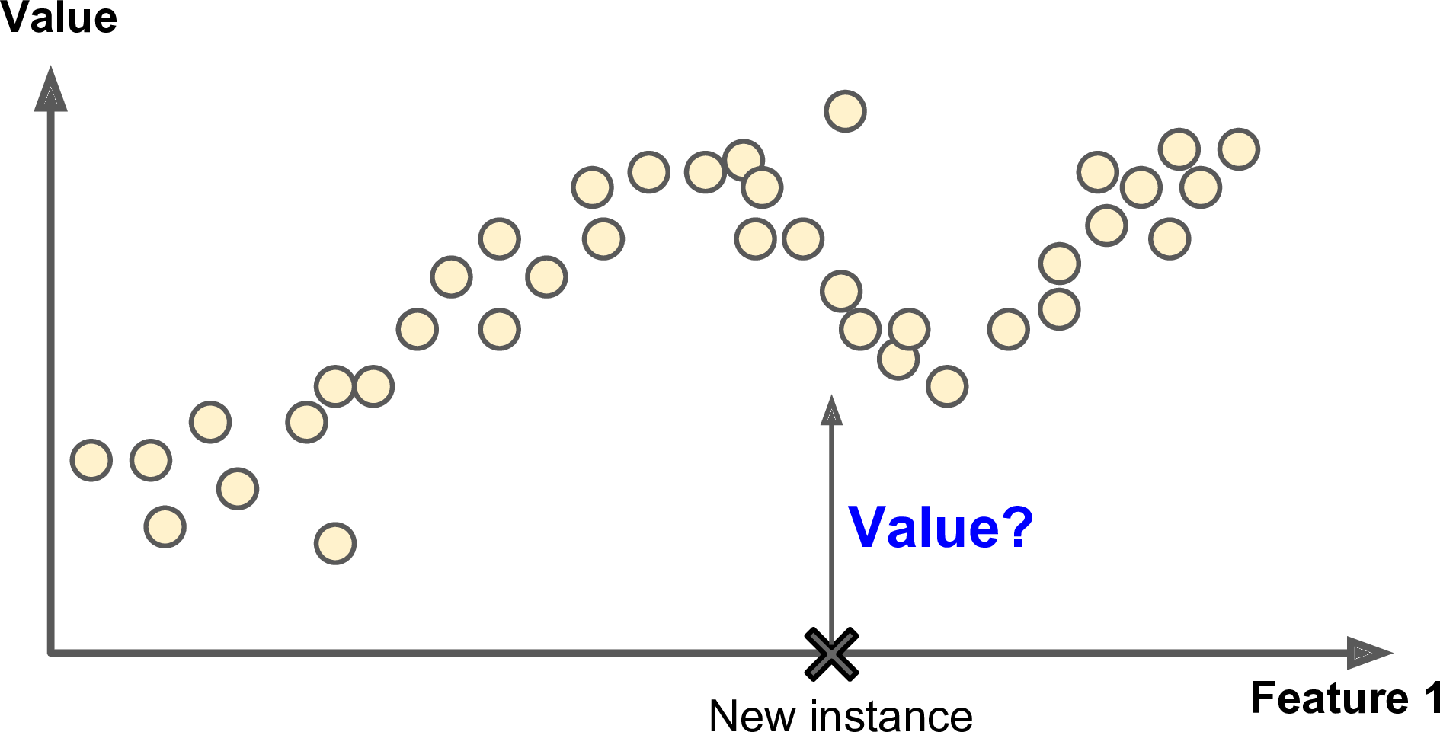
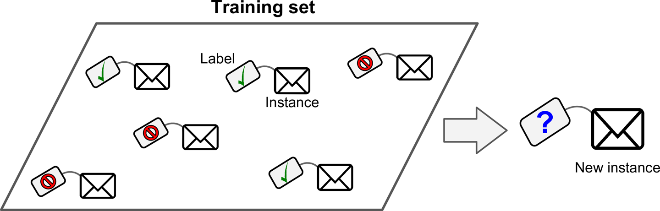
By examining the data, we can learn more about the problem itself and exploit the insights…

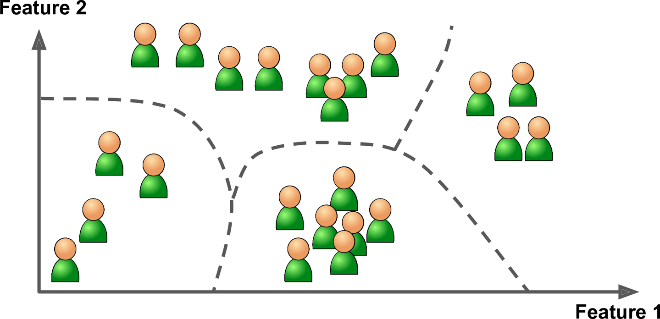


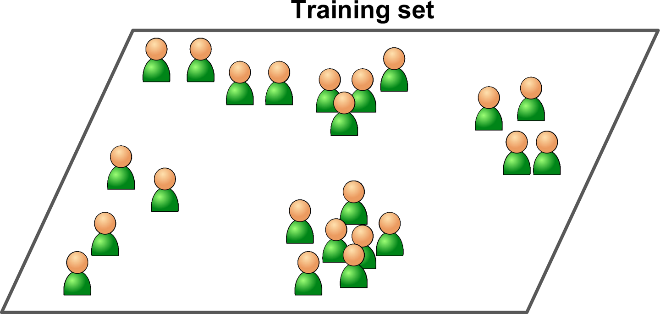
Ok! How do we *actually* do this ML thing?

Supervised vs. Unsupervised Learning  
Often depends on… Labeled vs. Un-Labeled data

Supervised learning uses labeled data to find relationships, allowing prediction…

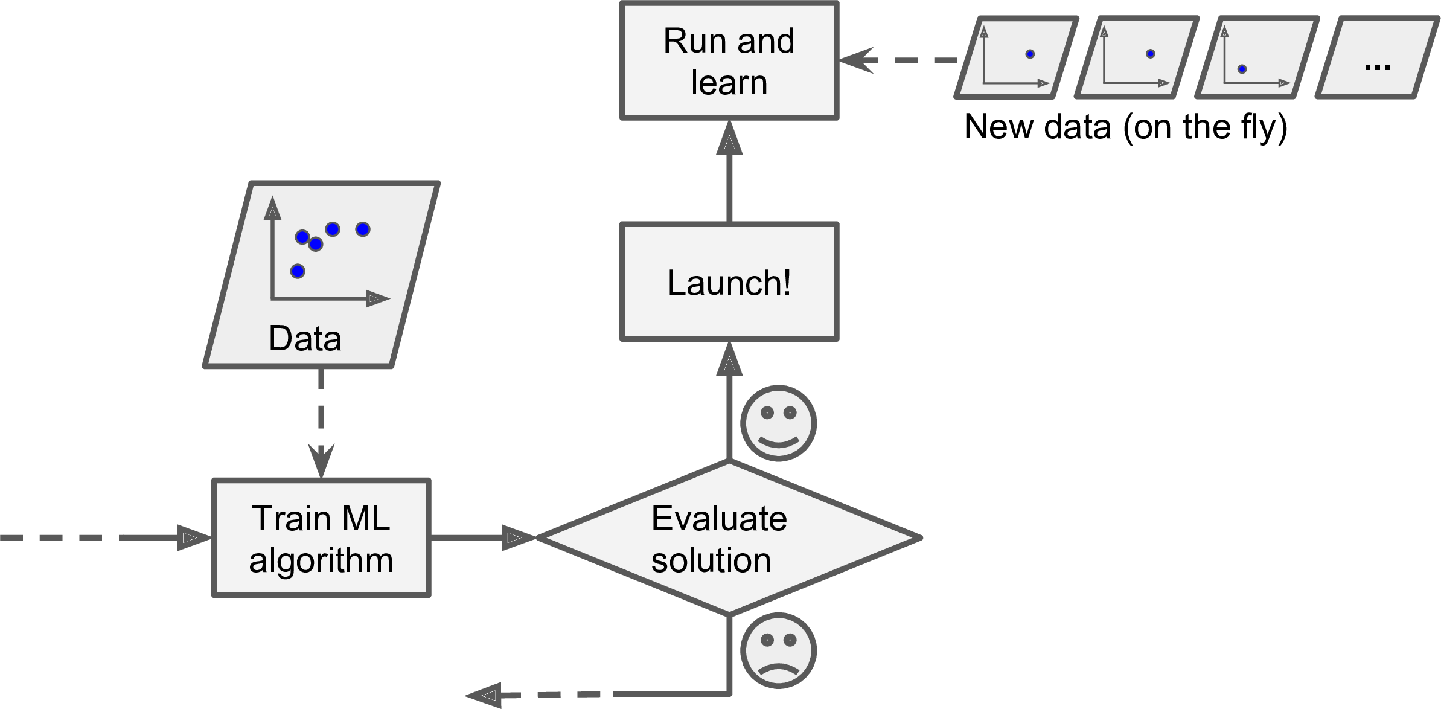


Unsupervised learning uses unlabeled data to uncover hidden patterns…

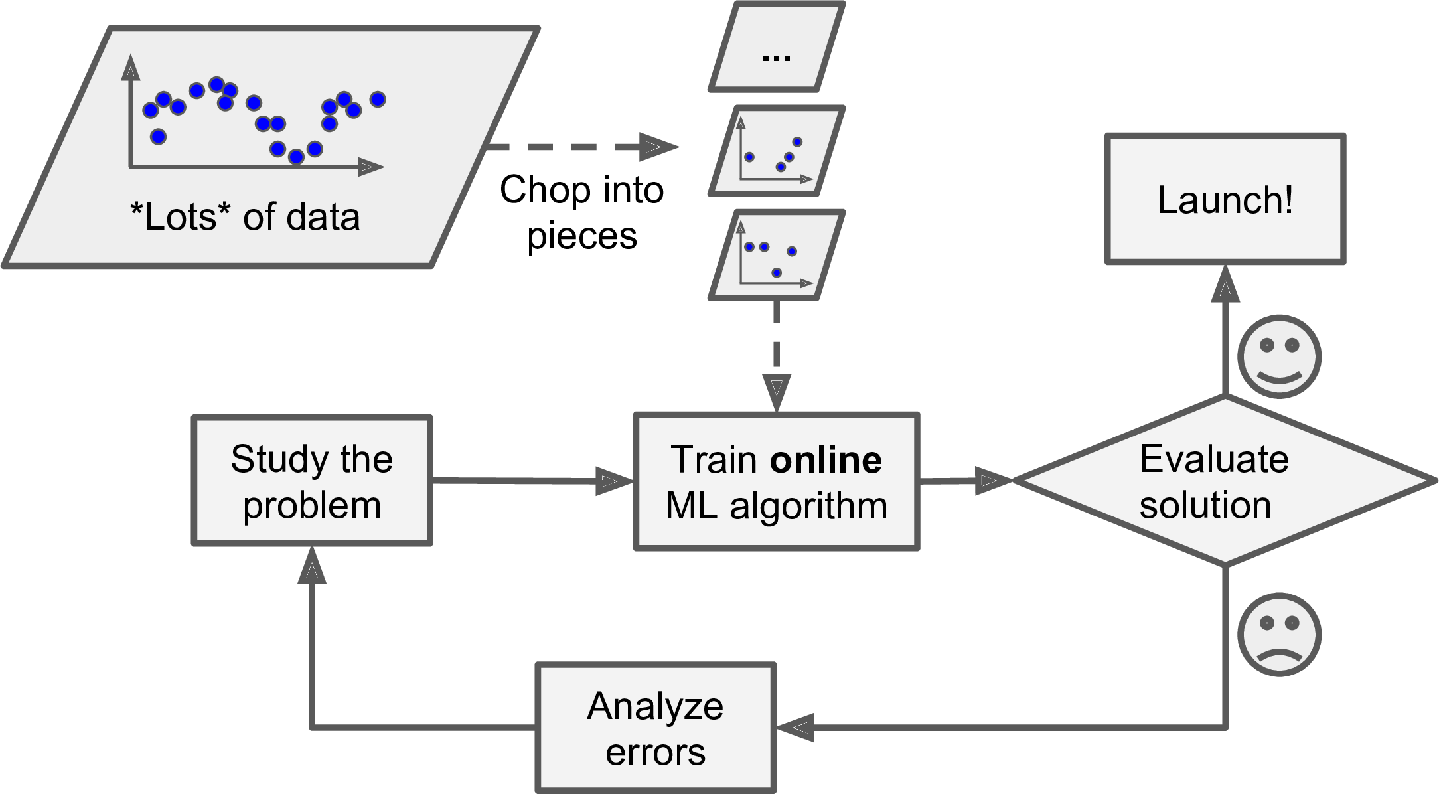


Online vs. Batch Learning

Online learns on the fly from incoming data…



Batch learns up front from significant and representative data…



Major high-level types of ML problems:

* Regression - outputs are real numbers or countable
* Classification - outputs are abstract or categorical
* Clustering - finds similarities among data, groups common data together into clusters
* Reinforcement Learning – independent agent optimizes actions based on rewards/penalties

Machine Learning work-flow…

1. Define problem
2. Gather and pre-process data
3. Select features
4. Choose model
5. Train model
6. Evaluate model
7. Iterate!
8. Use results

# Python Machine Learning

“Python is a programming language that lets you work more quickly and integrate your systems more effectively.” -- <https://www.python.org/>

Example from GeeksforGeeks  
<https://www.geeksforgeeks.org/python-program-to-check-whether-a-number-is-prime-or-not/>

# Python program to check if

# given number is prime or not

num **=** 11

# If given number is greater than 1

**if** num > 1:

    # Iterate from 2 to n / 2

**for** i **in** range(2, int(num**/**2)**+**1):

        # If num is divisible by any number between

        # 2 and n / 2, it is not prime

**if** (num **%** i) **==** 0:

            print(num, "is not a prime number")

**break**

**else**:

**print**(num, "is a prime number")

**else**:

    print(num, "is not a prime number")

Python is a real language, with all that entails 😊

Resources:

<https://www.python.org/doc/>

<https://docs.python.org/3/tutorial/index.html>

Jupyter Notebook

“The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.” -- <https://jupyter.org/>

Machine Learning open source libraries for Python:

* Pandas: The Pandas library provides users with the ability to handle large datasets. It provides tools for reading and writing data, cleaning and altering data, and so on.
* Numpy: The Numpy, or Numerical Python, library provides users with a powerful array of computing abilities. It tackles the problem of slow mathematical computations and allows users to perform huge calculations with the help of multi-dimensional arrays.
* Scipy: The Scipy library is used for scientific and technical computations. It works on Numpy’s multi-dimensional arrays.
* Scikit-Learn: The Scikit-Learn library consists of various features and methods that have specially been made to assist users in their machine learning requirements. It makes use of the Numpy library, specifically when it comes to array operations.
* TensorFlow: The TensorFlow library is an increasingly popular library that provides users with a large set of flexible and accessible tools for machine learning. You will be learning more about TensorFlow later on in this book.

-- Nikita Silaparasetty2

# Regression

“Task to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.) called predictors. … To train the system, you need to give it many examples of cars, including both their predictors and their labels (i.e., their prices).” -- Aurélien Géron1

By the way…

Attribute – data type (e.g. “mileage”)

Feature – attribute plus its value (e.g. “mileage = 15,000”)

Label – desired result (e.g. “price of car”)

Think of features as independent variables, X, and labels as dependent variable, Y  
ML is learning an algorithm for predicting Y from X, given a bunch of (X, Y) pairs

What are some example problems that could be solved by regression?

Important regression algorithms:

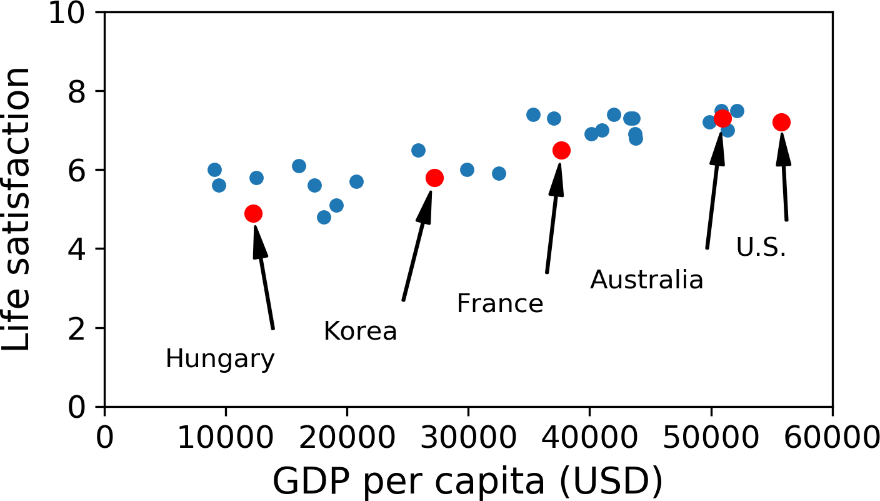
* k-Nearest Neighbors
* Linear Regression
* Logistic Regression
* Support Vector Machines (SVMs)
* Decision Trees and Random Forests
* Neural networks

# Does Money Make People Happy?

From Chapter 1 of the text (Aurélien Géron1)

For example, suppose you want to know if money makes people happy, so you download the Better Life Index data from the [OECD’s website](https://homl.info/4) and stats about gross domestic product (GDP) per capita from the [IMF’s website](https://homl.info/5).

Let’s visualize…

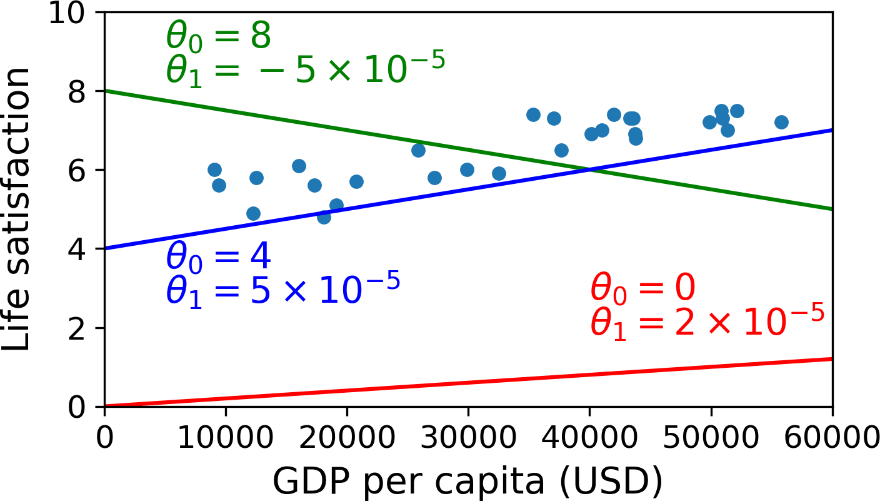


Is there a trend here?

A correlation perhaps?

Let’s try a linear model…

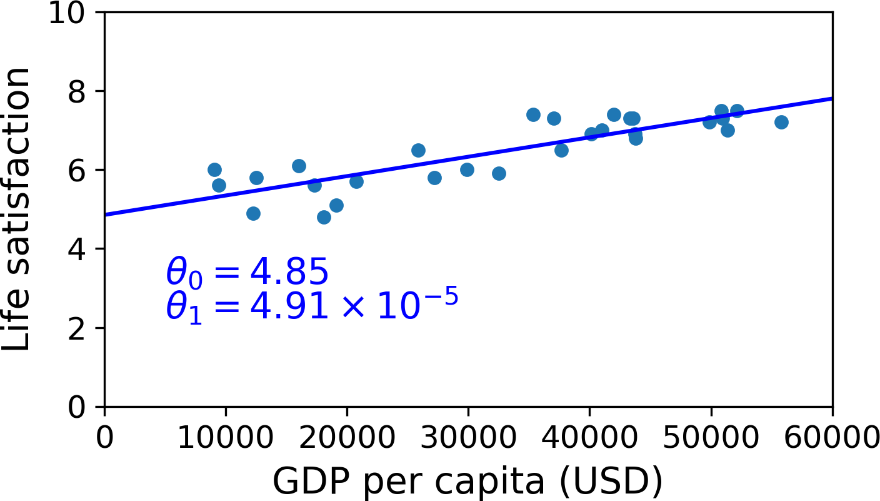
… with model parameters: θ0 and θ1



But what values should the model parameters be?

How do we find the values that give us the right model?

Training! Wait… What kind of training?



Using a Linear Regression algorithm (supervised, labeled, batch)

Now we can make predications…

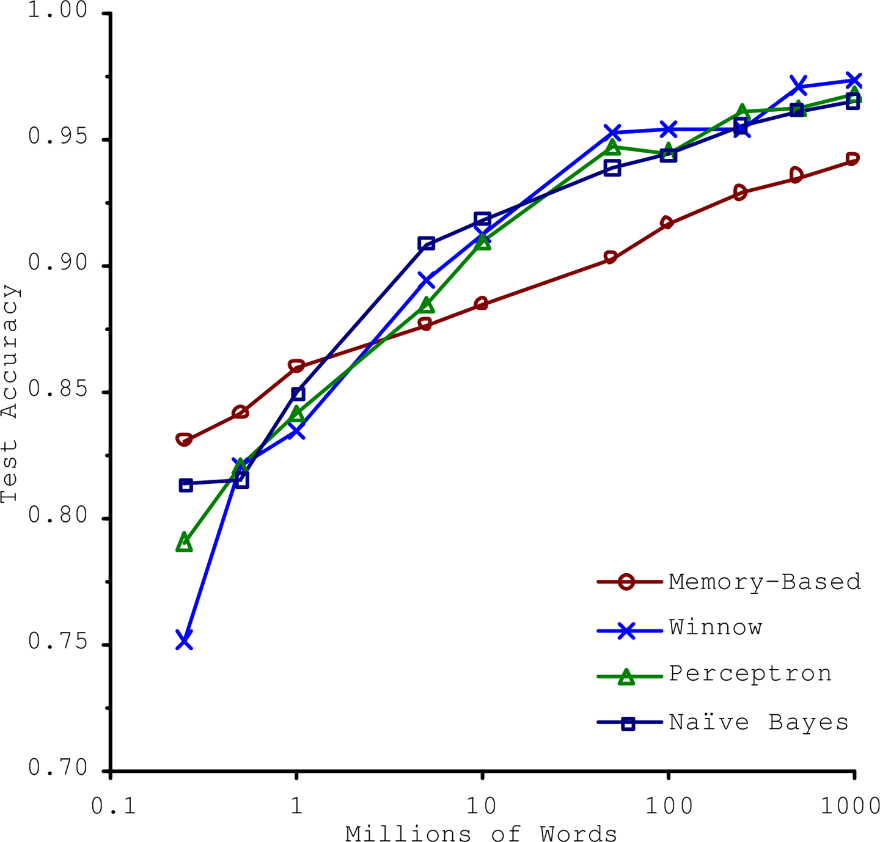
Cyprus’s GDP per capita: $22,587

Life satisfaction = 4.85 + 22,587 × 4.91 × 10-5 = 5.96

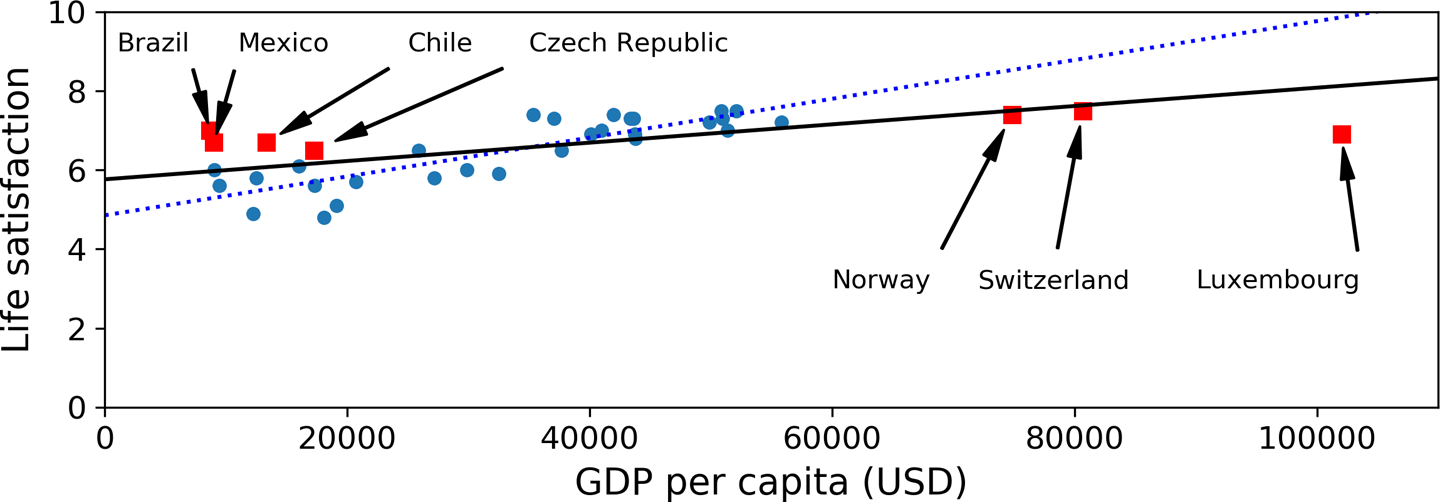
Check out how this is done in Jupyter Notebook: 01\_the\_machine\_learning\_landscape.ipynb

# ML Challenges

Insufficient Quantity of Training Data – takes thousands or millions of data points for most algorithms



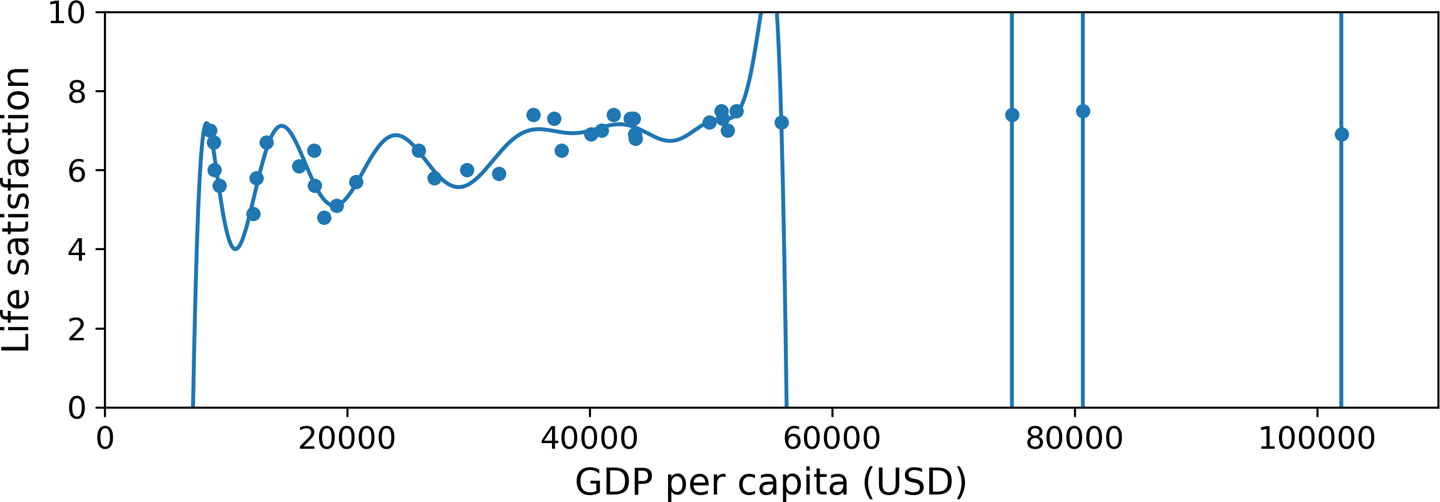
Nonrepresentative Training Data – leads to poor model fit to general data (low accuracy predictions)



Poor Quality Data – errors, outliers, noise… hard to detect patterns and inaccurate predictions

Irrelevant Features – cannot learn a non-existant relationship (low correlation)

Overfitting the Training Data – fits the training data ***really*** well, but doesn’t generalize



Underfitting the Training Data – model is too simple to represent the data, poor accuracy

# References

1. Hands-On Machine Learning with Sckit-Learn, Keras & TensorFlow, 2nd Edition, by Aurélien Géron
2. Machine Learning Concepts with Python and the Jupyter Notebook Environment: Using Tensorflow 2.0, by Nikita Silaparasetty
3. Introduction to Machine Learning 4th Edition, by Ethem Alpaydin