Intelligent Systems (IS)

Master's Degree in Informatics Engineering

End-to-End Machine Learning Project

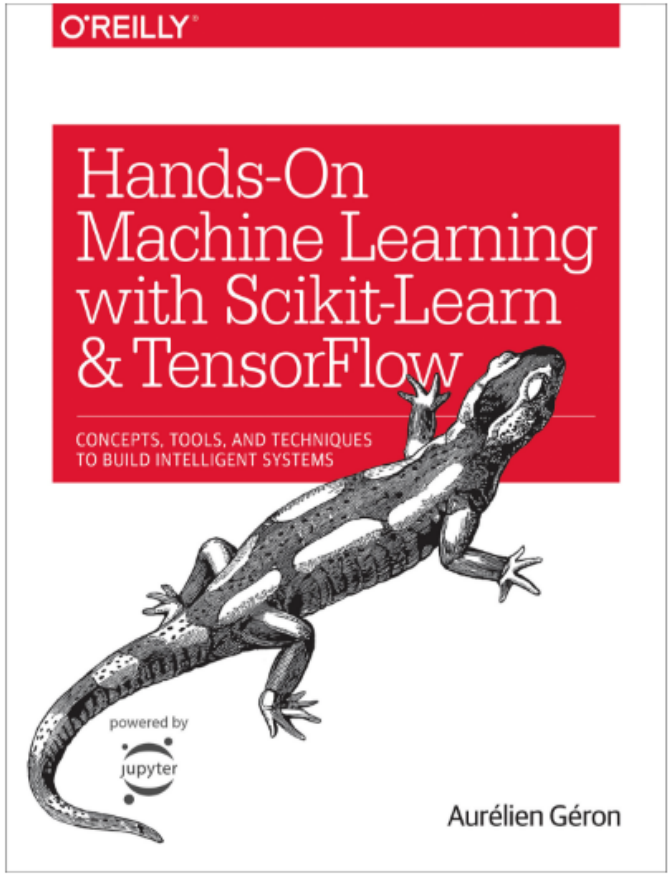
Carlos Ansótegui

Área: Ciencias de la Computación e Inteligencia Artificial (CCIA)
Departamento de Informática e Ingeniería Industrial
Escuela Universitaria Politécnica
Universidad de Lleida

September 1, 2017

Acknowledgements...

We will mostly follow:



Index

Machine Learning (ML) algorithms

- Supervised ML algorithms
- Unsupervised ML algorithms
- Reinforcement Learning (RL)

Supervised ML algorithms

Supervised algorithms:

Data: consists of a set of items, where each item is described by a set of attributes.

There is an special attribute called **the target or class attribute (or label)** which indicates the class the item belongs to.

Goal: to predict the value of the class attribute for items where it is missing.

Supervised ML algorithms

Goal: predict flower species

Sepal length	Sepal width	Petal length	Petal width	Species
5.0	*(missing)*	1.4	Small	Iris Setosa
5.0	3.5	1.6	Medium	Iris Setosa
(missing)	3.2	4.7	Big	Iris Versicolour
5.0	2.3	3.3	Medium	Iris Versicolour
6.3	3.3	6.0	Big	Iris Virginica

Supervised ML algorithms

Goal: predict median house value

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0

Supervised ML algorithms

ML phases: training and test phase.

During the **training phase**, the ML algorithm builds a predictor (or classifier) based on a dataset where the target or class attribute is known for every item.

These items are referenced as examples, or labelled data.

During the **test phase**, the predictor is used to classify unlabelled data.

Supervised ML algorithms

some supervised algorithms:

- Decision Trees and Random Forests
- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Neural Networks

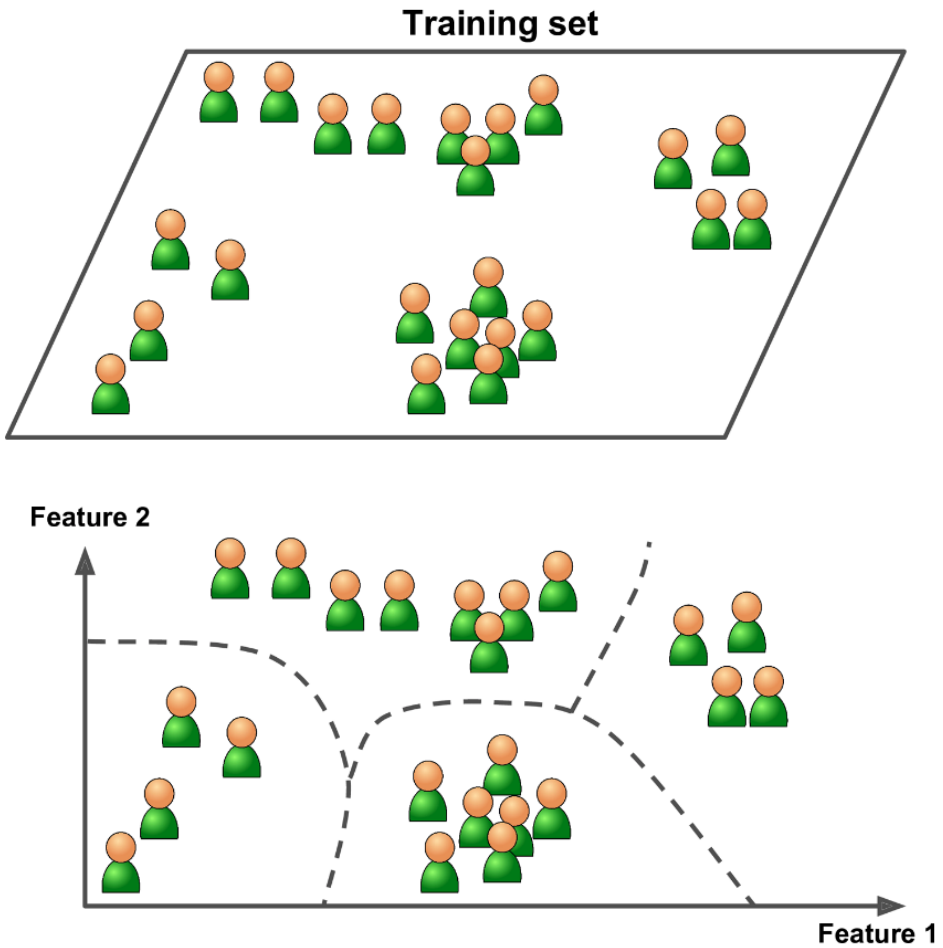
Unsupervised ML algorithms

Data: There are no examples, all the data is unlabelled.

Goal: discover the possible classes of items by clustering those that are more similar in terms of a **similarity function** applied to the attributes of the items.

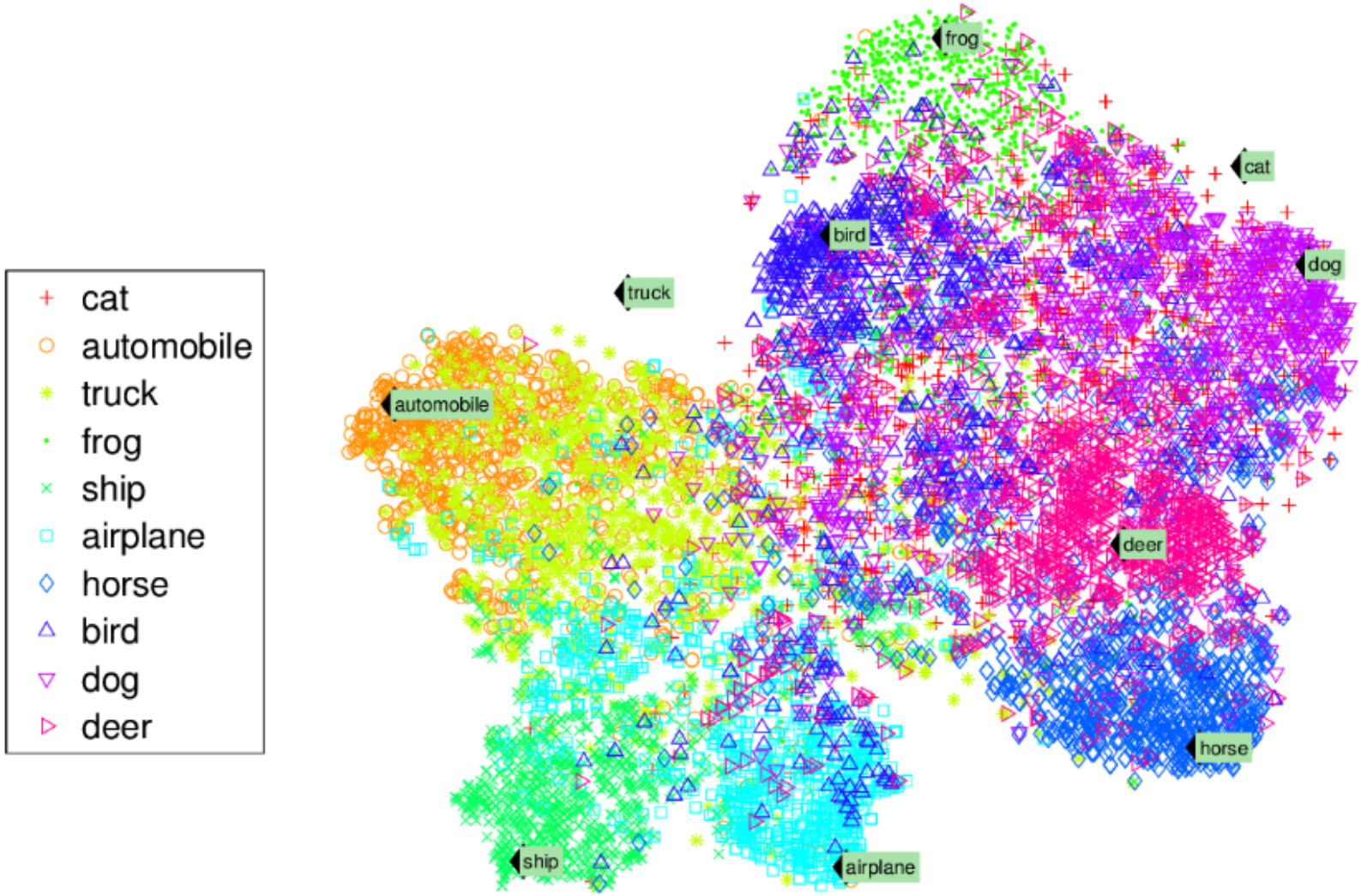
Unsupervised ML algorithms

Goal: discover clusters of users according to features 1 and 2.



Unsupervised ML algorithms

Goal: represent complex data in 2D or 3D



Unsupervised ML algorithms

Goal: anomaly detection

Examples: detecting unusual credit card transactions to prevent fraud, catching manufacturing defects, or automatically removing outliers from a dataset before feeding it to another learning algorithm.

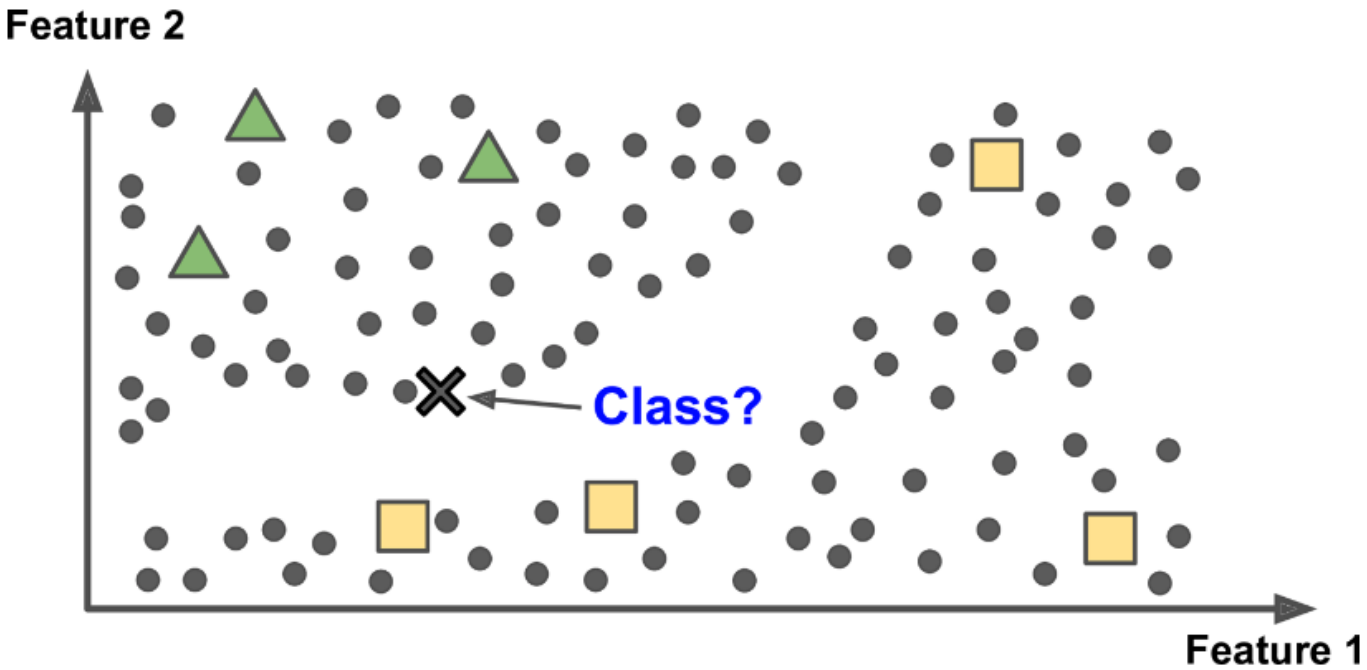
Unsupervised ML algorithms

some unsupervised algorithms:

- Clustering
  - k-Means
  - Hierarchical Cluster Analysis (HCA)
  - Expectation Maximization
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat

Semisupervised ML algorithms

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



Example of algorithm: Deep Belief Networks

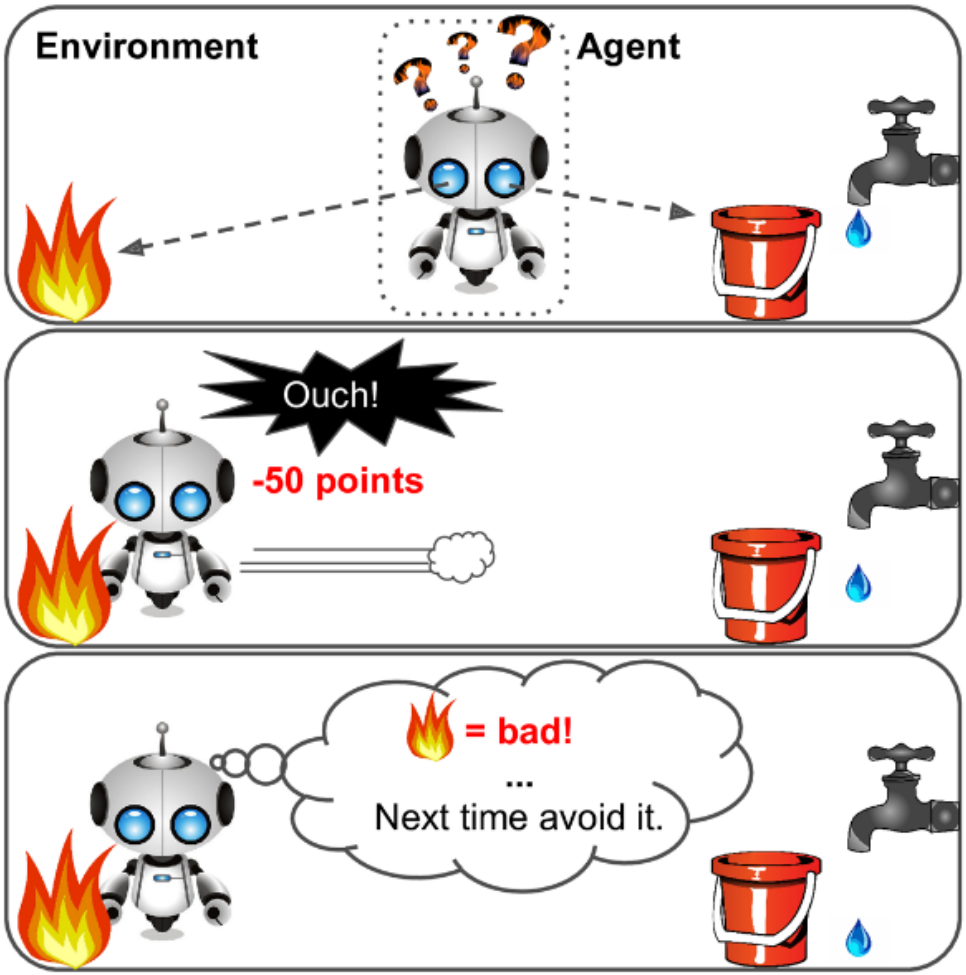
Reinforcement Learning (RL)

The learning system (agent), can observe the environment, select and perform actions, and get rewards in return (or penalties in the form of negative rewards, as in Figure 1-12).

Goal: Learn the best strategy (policy) to get the most reward over time.

A policy defines what action the agent should choose when it is in a given situation.

Reinforcement Learning (RL)



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

Reinforcement Learning (RL)

some RL algorithms:

- Value Iteration
- Policy Iteration
- Q-learning

Machine Learning Frameworks

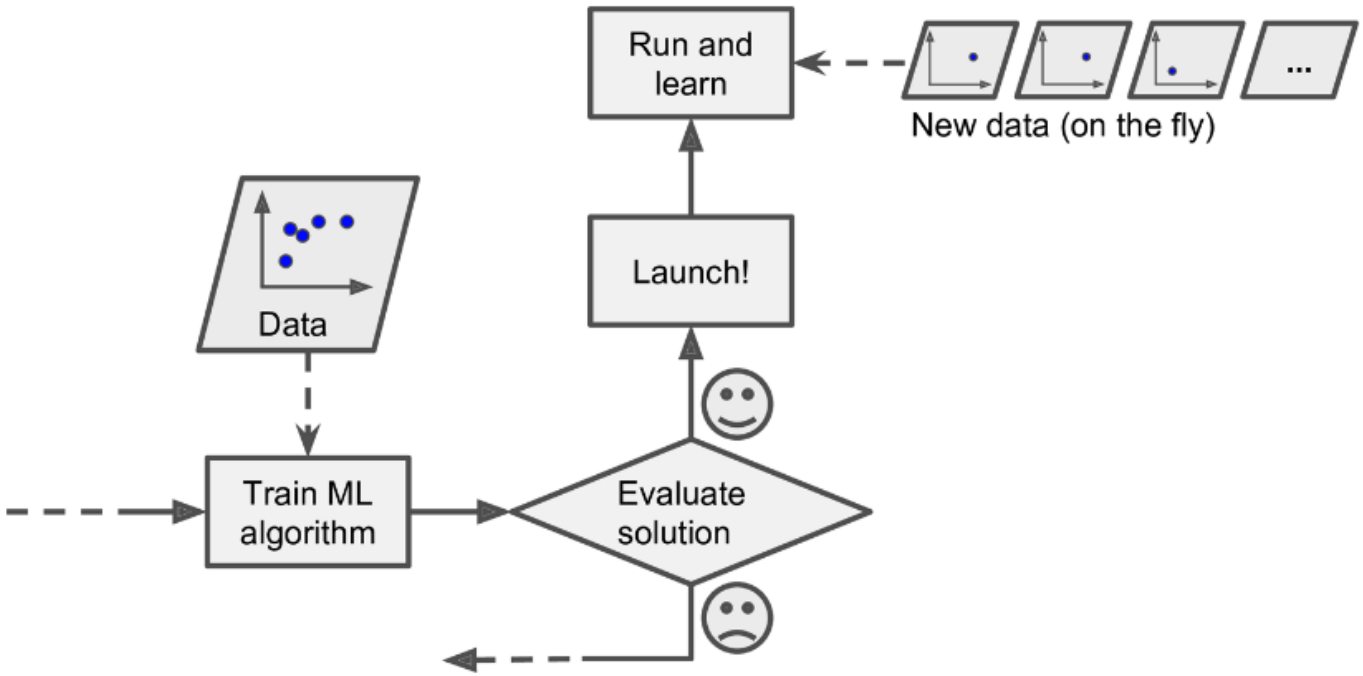
	BigML	Azure ML	scikit-learn	Weka
Price	Free but limited	Free but limitated	Free	Free
# parallel tasks	Limited	Limited	Unlimited	Unlimited
# parameters	Low	Medium	High	High
Open source code	No	No	Yes	Yes
# ML algorithms	Low	Medium	High	High
Programming language	*REST API*	*REST API*	Python	Java

BigML (https://bigml.com/)
Azure ML (https://azure.microsoft.com/en-us/)
scikit-learn (http://scikit-learn.org/)
Weka (https://www.cs.waikato.ac.nz/ml/weka/)

Batch and Online Learning

Batch learning: the system is incapable of learning incrementally.

Online learning: the system grows incrementally by feeding it data instances sequentally, either individually or by smallgroups called mini-batches.



Batch and Online Learning

Learning rate: how fast to adapt to changing data.

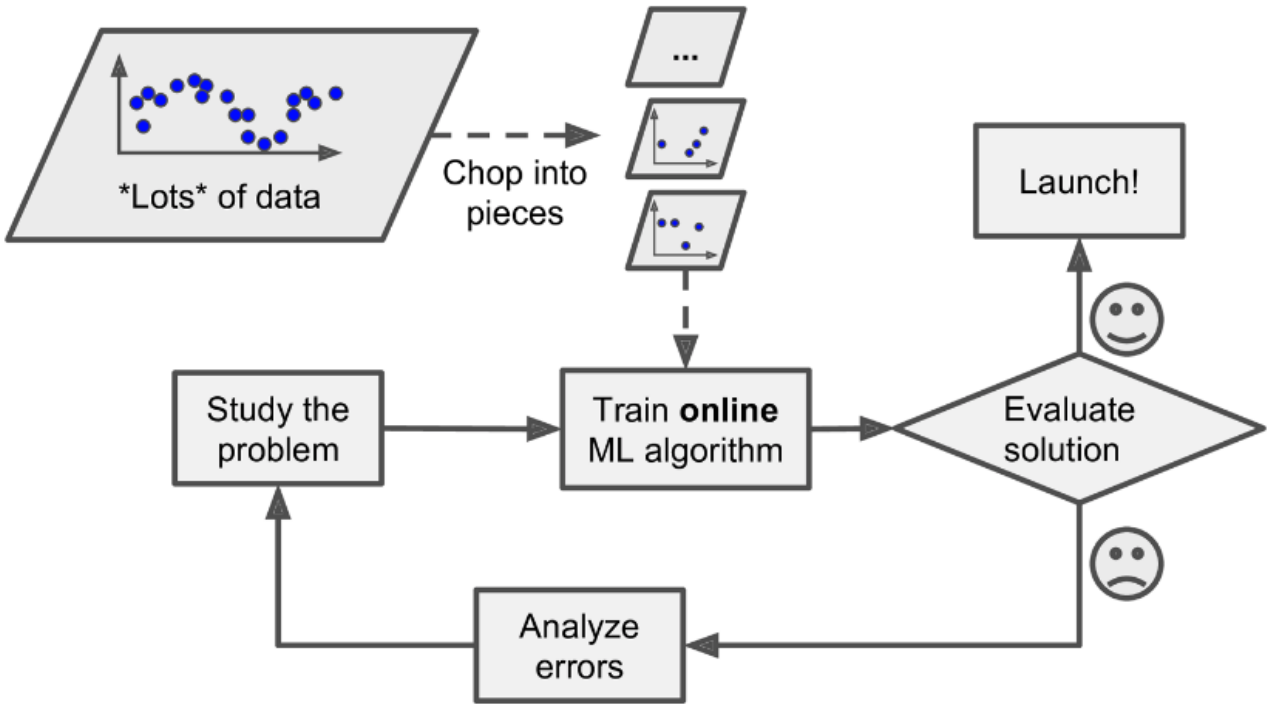
High learning rate: rapidly adapts to new data, but quickly forgets old data (spam filter only flags latest kind of spam ...).

Low learning rate: systems has more inertia, learns more slowly, less sensitive to noise.

Manage bad data: monitor your system closely, promptly switch learning off (possibly revert to a previous state). Idea: use anomaly detection algorithms.

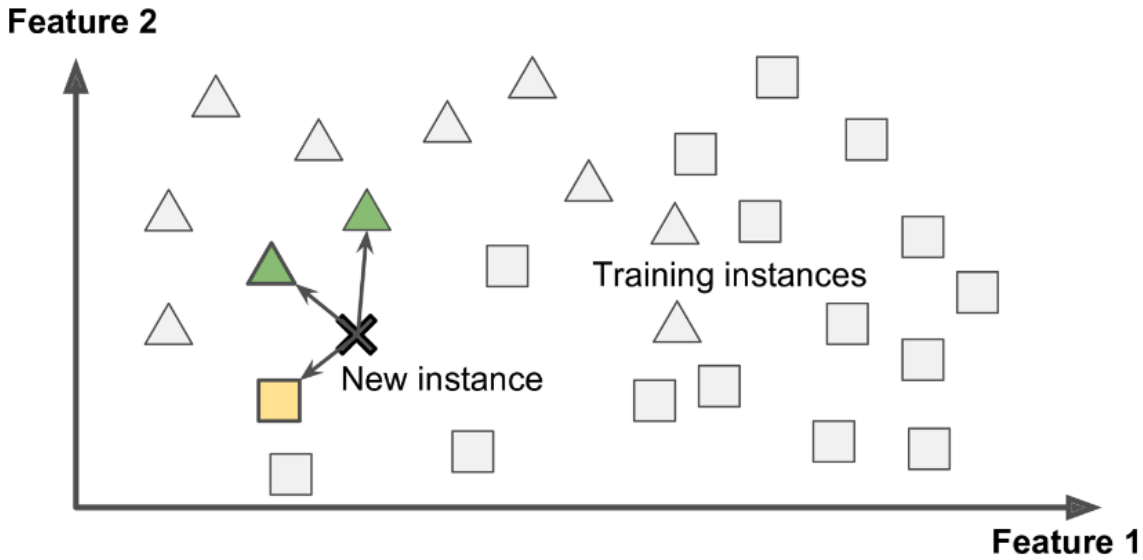
Batch and Online Learning

incremental learning: as online learning but it is an offline process to train systems on huge datasets



Instance-Based Versus Model-Based Learning

Instance based learning: instead of performing explicit generalization, compares new problem instances with instances seen in training.

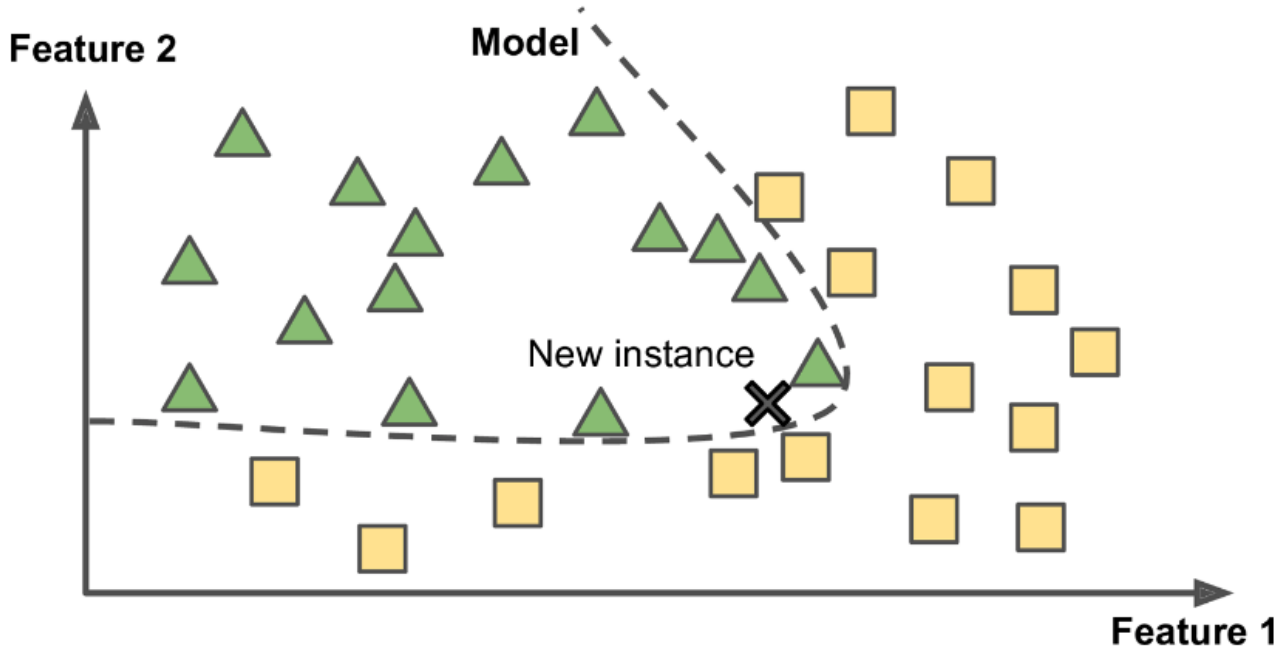


advantage: ability to adapt its model to previously unseen data. Instance-based learners simply store a new instance or throw an old instance away.

example algorithm: k-nearest neighbors

Instance-Based Versus Model-Based Learning

Model-based learning: makes a model of the examples, then uses the model to make predictions.



example algorithm: Decision trees, Neural networks, etc.

Model-Based Learning

Example: Does money make people happy?

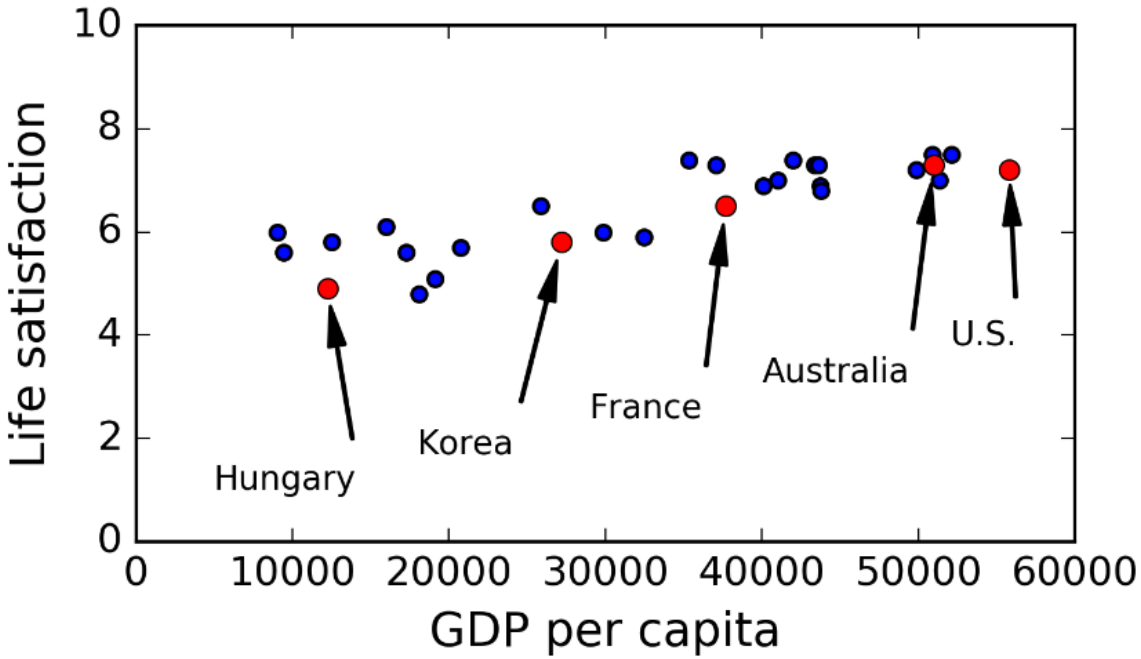
Lets load *Better Life Index* from OECD's website (https://goo.gl/OEht9W) and

GDP per capita from IMF's website (http://goo.gl/j1MSKe).

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

Model-Based Learning

Lets plot the data for a few random countries.

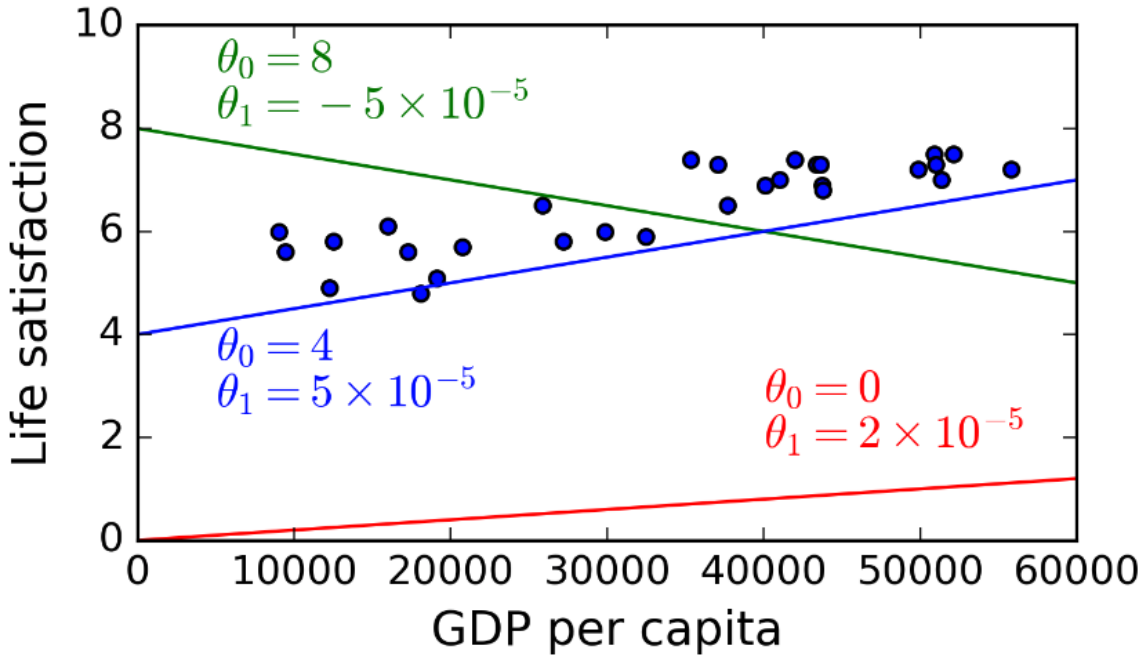


There seems to be a trend!

Lets model life satisfaction as a linear function of GDP per capita.

Model-Based Learning

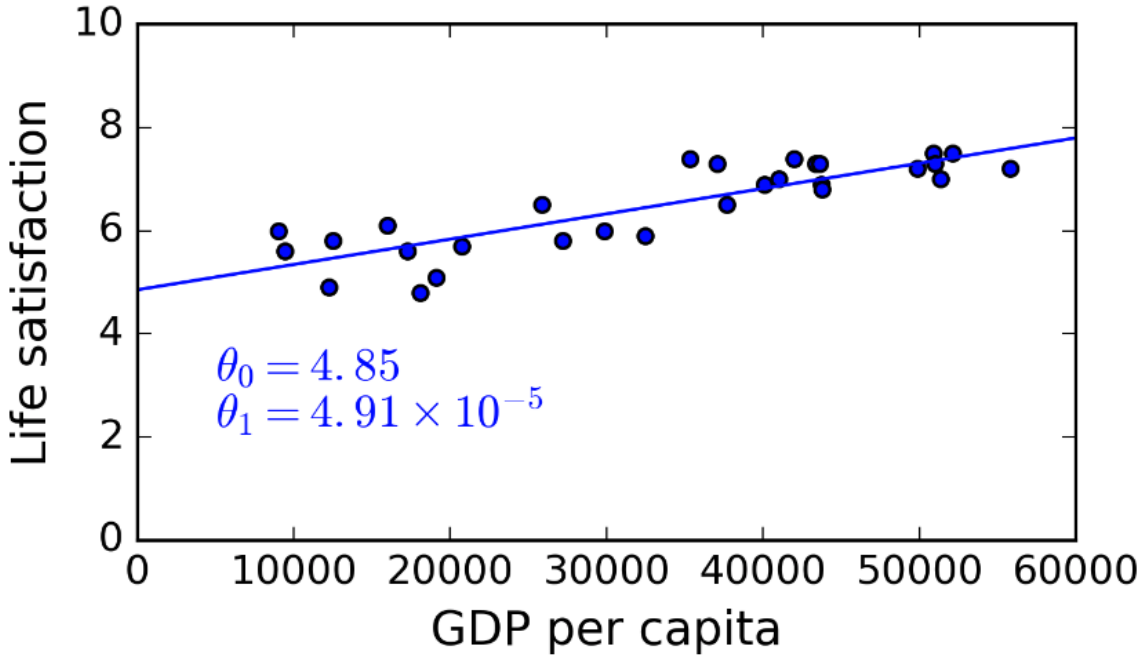
$$\textit{life\_satisfaction} = \theta_0 + \theta_1 \times \textit{GCD\_per\_capita}$$



by tweaking θ_0 and θ_1 your model can represent any linear function.

Model-Based Learning

Now, you can run a Linear Regression algorithm: it is fed with training examples and it finds the parameters that make the linear model fit best.



Model-Based Learning

Lets see some code ...

Download https://github.com/ageron/handson-ml.

Open jupyter.

Load 01_the_machine_learning_landscape.ipynb.

Index

End to End ML Project

During this class we will pretend you have been recently hired to develop a machine learning project:

(Hands-On Machine Learning with Scikit-Learn & TensorFlow. Aurélien Géron)

The steps we will follow are:

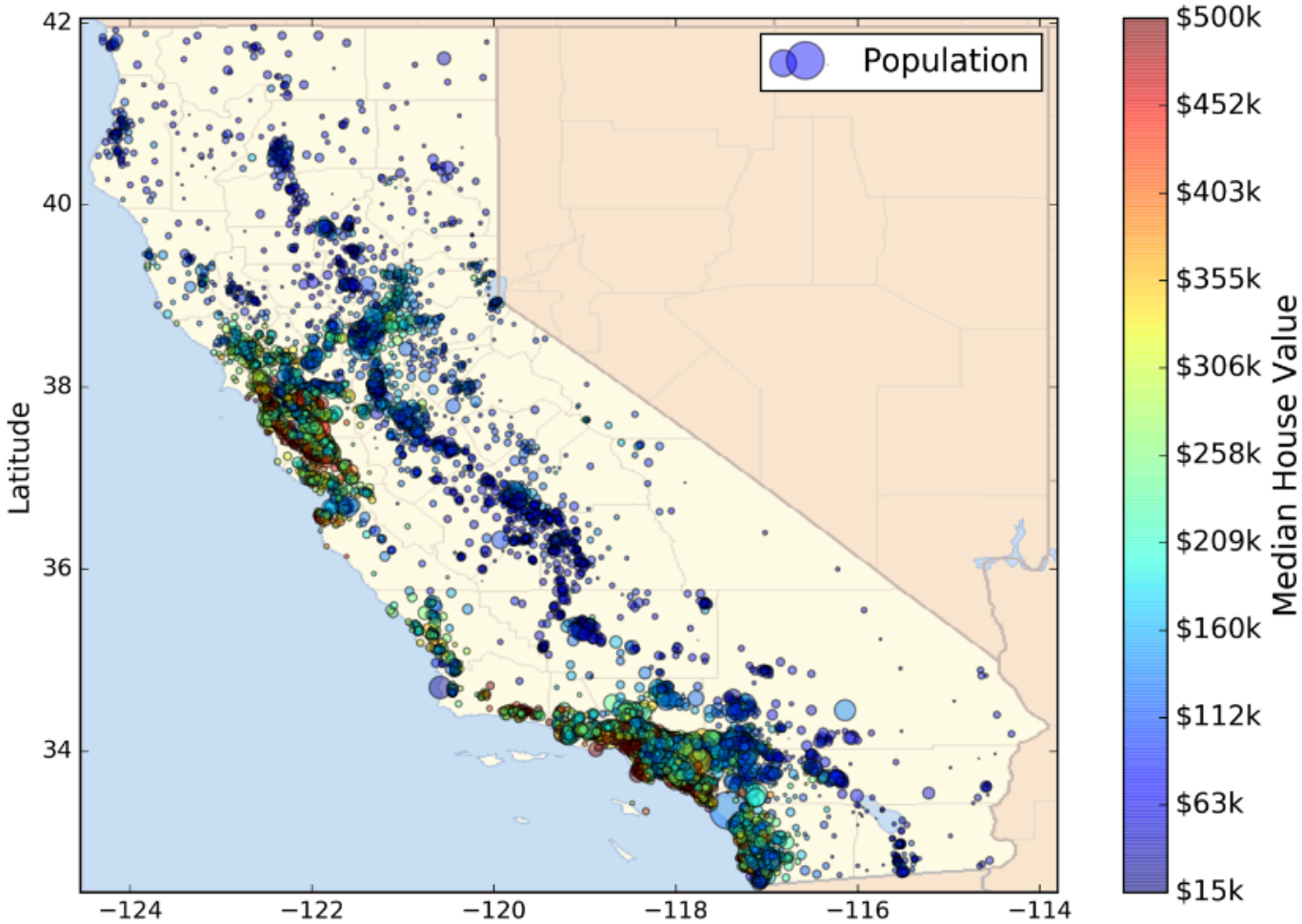
- Look at the big picture
- Get the data
- Discover and visualize the data to gain insights
- Select a model and train it
- Fine-tune your model
- Present your solution
- Launch, monitor, and maintain your system

Working with Real Data

- Popular open data repositories:
- UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/)
- Kaggle datasets (https://www.kaggle.com/datasets/)
- Amazonâs AWS datasets (https://aws.amazon.com/fr/datasets/)
- Meta portals (they list open data repositories)
- http://dataportals.org/
- http://opendatamonitor.eu/
- http://quandl.com/
- Other pages listing many popular open data repositories:
- Wikipedia list of Machine Learning datasets (https://goo.gl/SJHN2k)
- Quora.com question (http://goo.gl/zDR78y)
- Datasets subreddit (https://www.reddit.com/r/datasets)

Working with Real Data

In this class we will use the California Housing Prices from Statlib repository (data from the 1990 California census).



Look at the Big Picture

Goal: learn from the data a model that is able to predict median housing price in any other district

Data source: California census data

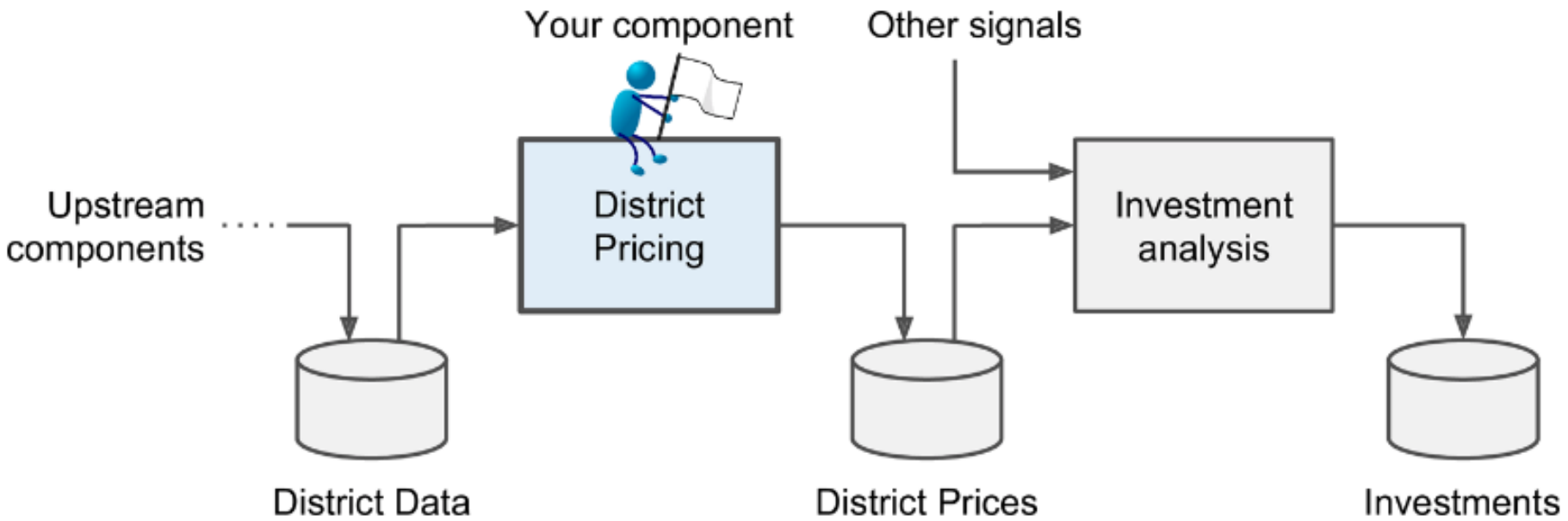
Data: metrics such as: population, median income, median housing price, and so on for each block group in California.

Block group (districts): smallest geographical unit (from 300 to 3000 people)
What now?:

1. Frame the problem
2. Select a performance measure
3. Check the assumptions

Frame the problem

Which is the business final objective?
How does the company expect to use and benefit from the model?



Frame the problem

What algorithms to use?

- supervised, unsupervised or reinforcement learning?
- classification or regression?
- batch learning or online learning?

What performance measure to evaluate the model?

How much effort to spend tweaking it?

How does the current solution look like?

Frame the problem

What algorithms to use?

- **supervised**, unsupervised or reinforcement learning?
- classification or **regression**?
- **batch learning** or online learning?

What performance measure to evaluate the model?

How much effort to spend tweaking it?

How does the current solution look like?

Select a performance measure

Typical measure for regression:

Root Mean Square Error (RMSE):

Measures the standard deviation of the errors the system makes in its predictions.

$$RMSE(X, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2}$$

Select a performance measure

Notation:

- m number of instances in the dataset
- $x^{(i)}$ is a vector of all the feature values (excluding the label) of the i^{th} instance in the dataset
- $y^{(i)}$ is the label of the of the i^{th} instance in the dataset

Example:

If the first district in the dataset is located at longitude $-118.29°$, latitude $33.91°$, and it has 1416 inhabitants with a median income of \$38372 and the median house value is \$156400, then:

$$\mathbf{x}^{(1)} = \begin{pmatrix} -118.29 \\ 33.91 \\ 1416 \\ 38372 \end{pmatrix} \qquad \text{and, } y^{(1)} = 156400$$

Select a performance measure

Notation:

- X is a matrix containing all the feature values (excluding labels) of all instances in the dataset.

Example:

$$x^{(1)} = \begin{pmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m-1)})^T \\ (x^{(m)})^T \end{pmatrix} = \begin{pmatrix} -118.29 & 33.91 & 1416 & 38372 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Select a performance measure

- h is the system's prediction function (hypothesis), i.e., $\hat{y} = h(x^{(i)})$

Example:

If your system predicts that the median housing price in the first district is 158400 then $\hat{y}^{(1)} = h(x^{(1)}) = 158400$. the prediction error is $\hat{y}^{(1)} - y^{(1)} = 2000$

- $RMSE(X, h)$ is the cost function measured on the set of examples using your hypothesis h.

Check the assumptions

List and verify the assumptions that were made so far.

Example: we have assumed that district prices (output of our system) will be fed into a downstream ML system.

What happens if the prices are converted into categories (cheap, medium, expensive, etc)?.

Check the assumptions

List and verify the assumptions that were made so far.

Example: we have assumed that district prices (output of our system) will be fed into a downstream ML system.

What happens if the prices are converted into categories (cheap, medium, expensive, etc)?.

We should have framed our problem as a classification task!.

Lets get hands dirty!

Lets see some code ...

Download https://github.com/ageron/handson-ml.

Open jupyter.

Load 02_end_to_end_machine_learning_project.ipynb