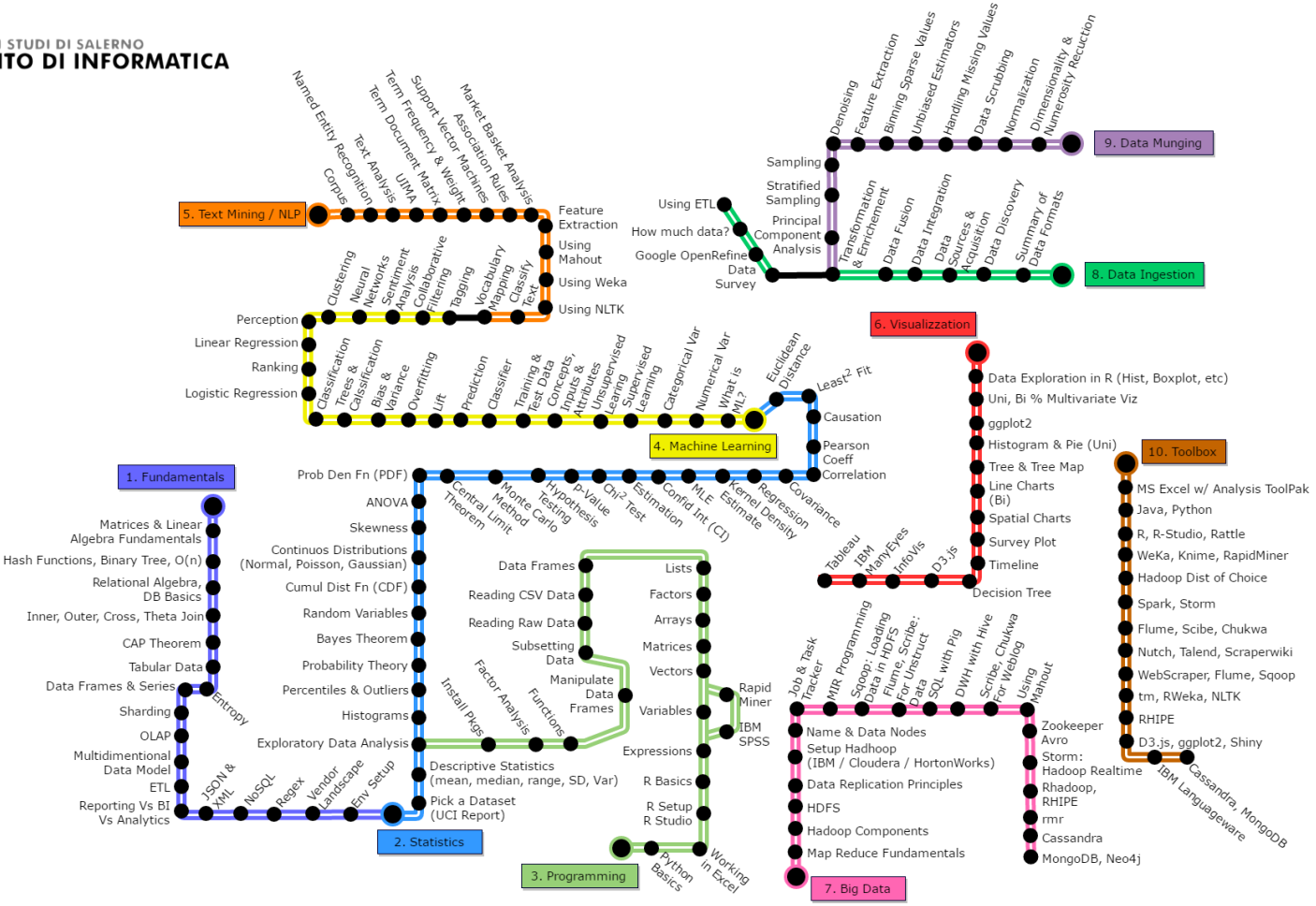
UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA



Fondamenti di Data Science e Machine Learning

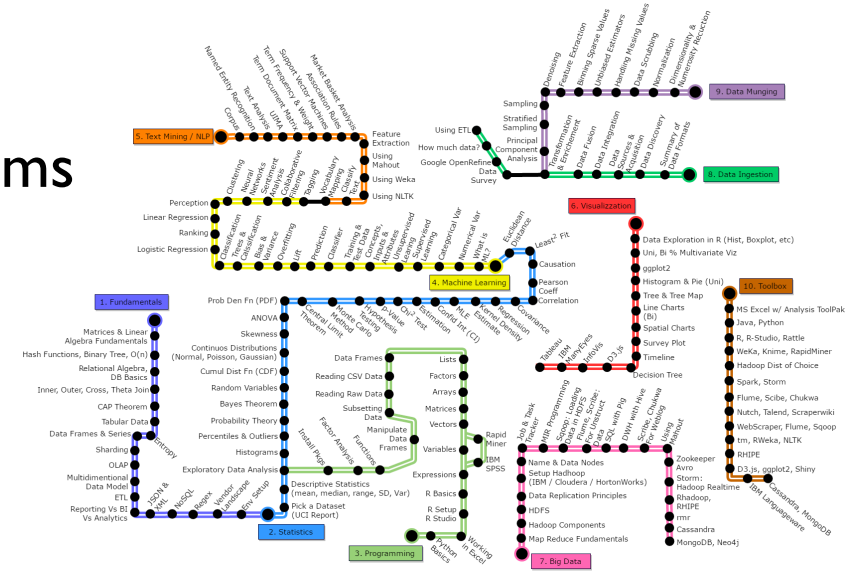
Introduction to Machine Learning

Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.

Prof. Giuseppe Polese, aa 2024-25

Overview

1. **Some History**
2. Machine Learning Examples
3. Goals of Machine Learning
4. Types of Machine Learning Systems
5. Problems in Machine Learning
6. Application scenarios
7. Summary



Some History

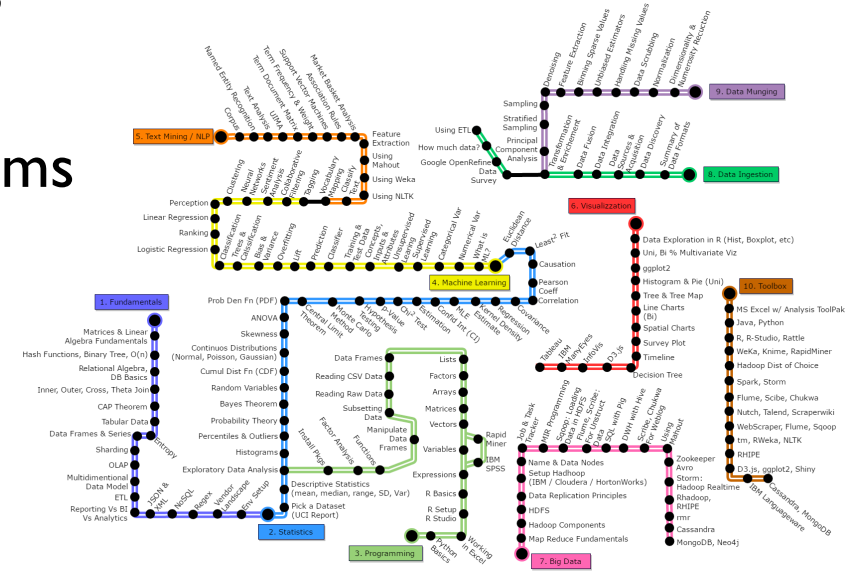
- Machine Learning has been initially used for in some specialized applications, such as Optical character Recognition (OCR).
- But the first widespread ML application arrived in the 1990s: the spam filter.
- It has been followed by many more ML applications used in everyday life, from recommendations systems to sentiment analysis.

What is Machine Learning?

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- *More generally*: Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. —*Arthur Samuel, 1959*
- A more engineering-oriented definition: 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*

Overview

1. Some History
2. **Machine Learning Examples**
3. Goals of Machine Learning
4. Types of Machine Learning Systems
5. Problems in Machine Learning
6. Application scenarios
7. Summary



Machine Learning Examples?

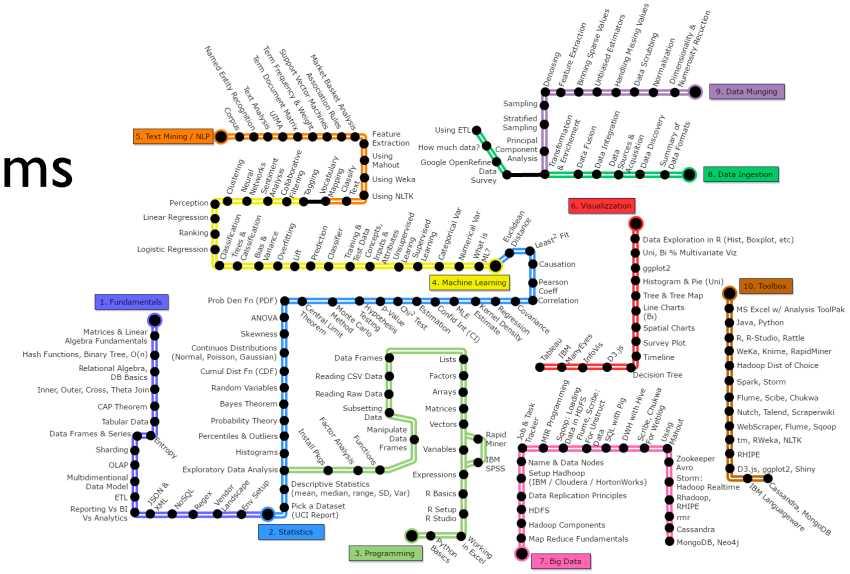
- A spam filter is a Machine Learning program that can learn to recognize spam from examples of spam emails flagged by users, and examples of nonspam ones (also called "ham").
- The examples from which the system learns are called the training set.
- Here, the task T is to classify new emails, the experience E is the training data, and the performance measure P can be for example the ratio of correctly classified emails.

Why Use Machine Learning?

- If you want to write a spam filter by using traditional programming techniques:
  - First you look for patterns of words and phrases (such as "4U," "credit card," …) appearing in many email subjects, sender's name, and so on.
  - Then, you write a detection algorithm for each of the detected patterns, classifying as spam emails containing a given number of these patterns.
  - Finally, you test your program, and repeat first two steps until it is good enough.

Overview

1. Some History
2. Machine Learning Examples
3. **Goals of Machine Learning**
4. Types of Machine Learning Systems
5. Problems in Machine Learning
6. Application scenarios
7. Summary



ML vs Traditional Approach?

- The program for spam detection will likely become a long and hard to maintain list of complex rules.
- Viceversa, an ML based spam filter detects unusually frequent patterns of words in the spam examples compared to the ham examples, so it automatically learns words and phrases that are predictors of spam.
- The program is shorter and easier to maintain, and continues to work if spammers change patterns if they realize that their emails are blocked (viceversa, traditional programs need be continuously changed).

ML use to reduce complexity

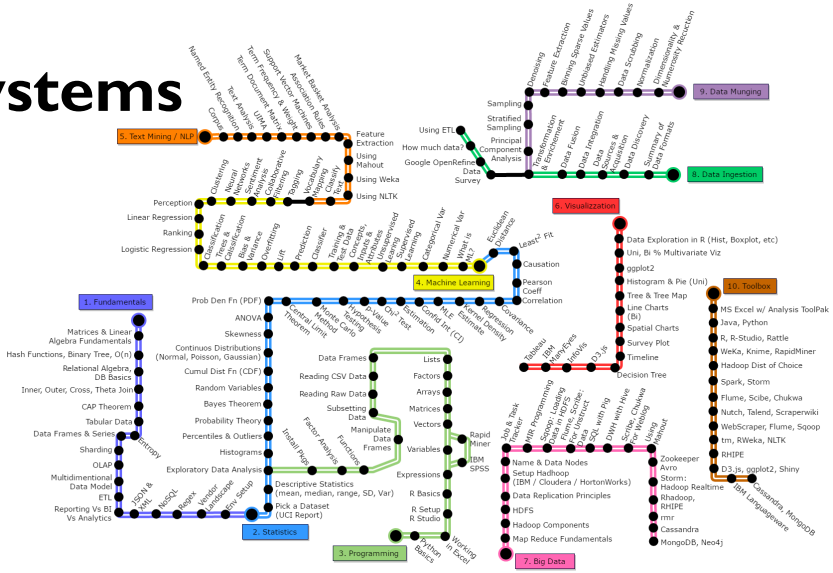
- ML can also be used for problems in which traditional approaches provide complex or no algorithm.
- For example, a speech recognition program might measure pitch sounds to distinguish two words.
- Obviously, this technique will not scale to thousands of words spoken by millions of very different people in noisy environments and in dozens of languages.
- The best solution is to write an algorithm that learns from example recordings for each word.

Learning from ML

- ML algorithms can be inspected to see what they have learned.
- For instance, once the spam filter has been completely trained, it can easily be inspected to verify the list of words and their combinations that it considers as predictors of spam.
- This can reveal unknown correlations that can lead to a better understanding of the problem.

Overview

1. Some History
2. Machine Learning Examples
3. Goals of Machine Learning
4. **Types of Machine Learning Systems**
5. Problems in Machine Learning
6. Application scenarios
7. Summary



Types of ML Systems

- ML systems can be classified based on:
  - Whether or not they are trained with human supervision (supervised, unsupervised, semisupervised, and Reinforcement Learning);
  - Whether or not they can learn incrementally (online versus batch learning);
  - Whether they learn by comparing new data to known data, or by detecting patterns in the training data to build a predictive model (instance-based versus model-based learning).

Types of ML Systems (2)

- The classification criteria of ML systems are not mutually exclusive.
- Thus, a typical ML system combines the characteristics of several classes.
- For example, we can have an online (e.g. on the fly), model-based, supervised learning system, and so on.

Supervised/Unsupervised ML

- In this class of ML systems there are four major categories:
  - Supervised learning;
  - Unsupervised learning;
  - Semisupervised learning;
  - Reinforcement Learning.
- Depending on the amount and type of supervision they get during training.

Supervised Learning

- In supervised learning, the training data are labeled with the desired solution.
- Classification is a typical supervised learning task.
- The spam filter is a good example: it can be trained with many example emails either labeled as spam or ham, and it must learn how to classify new emails.

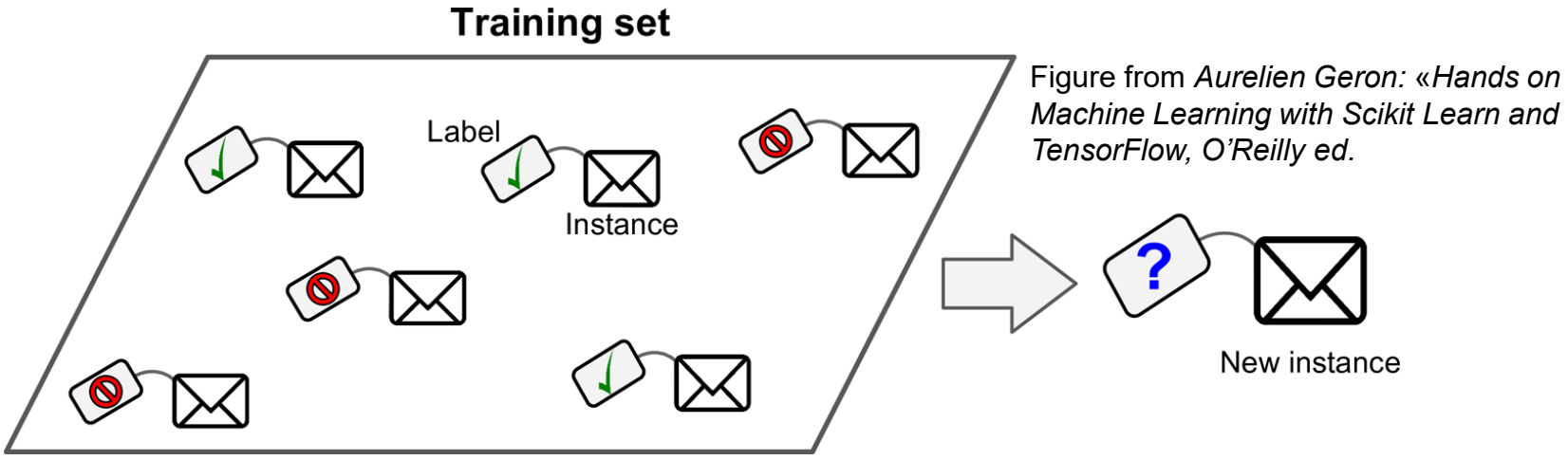


Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Supervised Learning: *Regression*

- Regression is another typical supervised learning task.
- With respect to Classification, the task here is to predict a numeric value instead of a class.
- For example, if the training data are features of cars, such as mileage, age, brand, and so on (also called predictors.), each car labeled with its price, a regression task might be to predict the price of a new car given its predictors.

Classification & Regression

- Classification and Regression can also be combined.
- A regression algorithm can also be used for classification, and viceversa.
- For example, Logistic Regression is also used for classification, since it can return a value indicating the probability of belonging to a given class.

Supervised Learning Algorithms

Main supervised learning algorithms are:

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks (although they can also be unsupervised and semisupervised)

Unsupervised Learning

- In unsupervised learning the training data is unlabeled.
- The system tries to learn without a teacher.

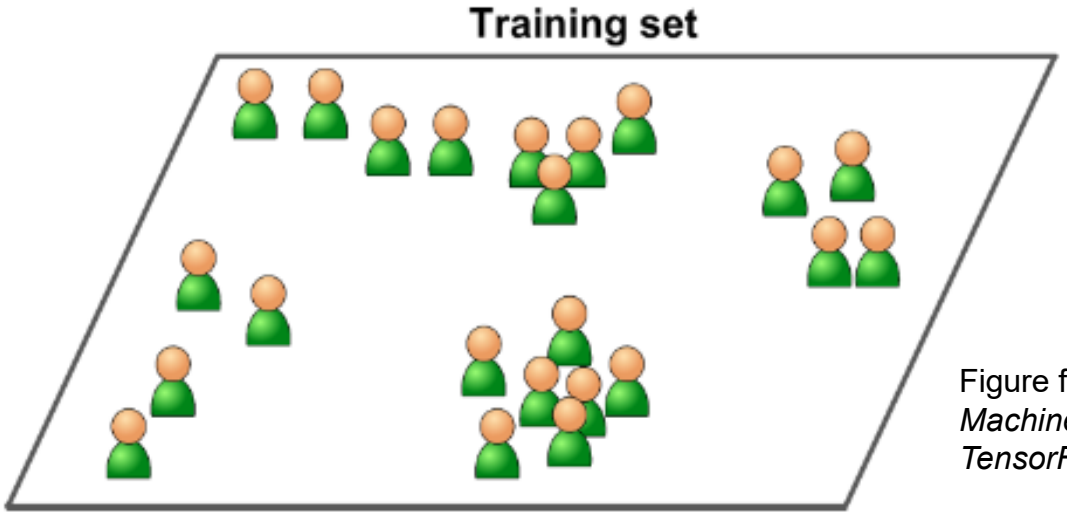


Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Unsupervised Learning Algorithms

Main unsupervised learning algorithms are:

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

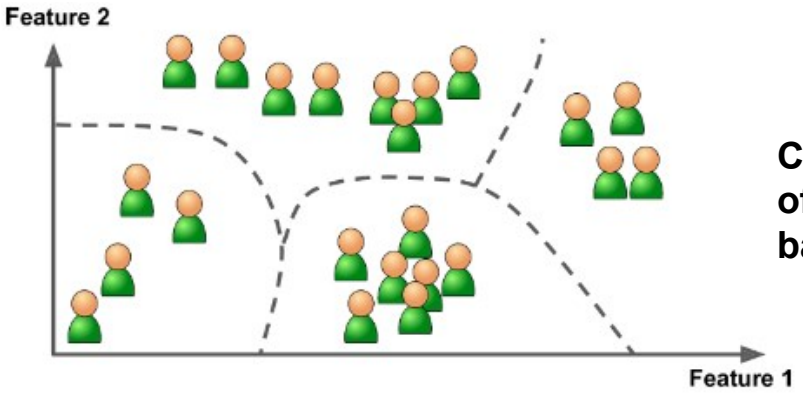
Unsupervised Learning Example

- A utility company owns a lot of data about its customers.
- It might want to use a clustering algorithm to detect groups of similar customers.
- The algorithm will never be told which group a customer belongs to: it finds those connections without help. For example, it might notice that 40% of customers are males, and in winter consume gas from 6 pm to 9 pm, whereas 15% are housewives who consume gas from 3 pm to 9 pm and during the weekends, and so on.
- A hierarchical clustering algorithm may also subdivide each group into smaller groups, which might help the company customize the communications for each group.

Learning through Visualization

- Visualization algorithms are also examples of unsupervised learning algorithms.
- They can be fed with a lot of complex and unlabeled data, and output a 2D or 3D representation of them, so that a user can probably identify previously unknown patterns.
- They try to preserve the original structure of data, trying to keep separate clusters in the input space from overlapping in the visualization.

Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*



Clustering customers of a utility company based on two features

Unsupervised Anomaly Detection

- An important unsupervised learning task is anomaly detection.
- For example, it can be used to detect unusual credit card transactions to prevent fraud, to catch manufacturing defects, etc.
- The system is trained with normal instances, so for a new instance it can tell whether it looks like a normal one or if it is likely to be an anomaly.

Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Association Rule Learning

- Association rule learning (a.k.a. Association rule mining) is another common unsupervised task.
- The goal here is to dig into large amounts of data and discover interesting relations between attributes.
- For example, running an association rule learning algorithm on the sales of a supermarket may reveal that people who purchase milk and eggs also tend to buy flower.
- Thus, it might be convenient to place these items close to each other. Or we might decide which products should be advertised or promoted to increase flower sales.

Semisupervised Learning

- Semisupervised learning algorithms deal with partially labeled training data (usually a lot of unlabeled data and few labeled ones).
- Most semisupervised learning algorithms are combinations of unsupervised and supervised algorithms.
- For example, deep belief networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on top of one another. RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

Semisupervised Learning Example

- Photo-hosting services, such as Google Photos, are good examples of semisupervised learning.
- When uploading all of a family photos to the service, it might use unsupervised clustering to automatically recognize 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.
- In the supervised part the algorithm needs a supervisor to assign a label per person, so as to give a name to all persons in the photo album, which will be useful for searching photos.

Reinforcement Learning

- In reinforcement Learning an agent observes an environment, selects and performs actions, gaining rewards or penalties.
- It must learn the best strategy, called a policy, aiming to get the most reward over time.
- A policy defines what action the agent should choose when it is in a given situation.
- Many robots implement reinforcement Learning to learn how to walk.

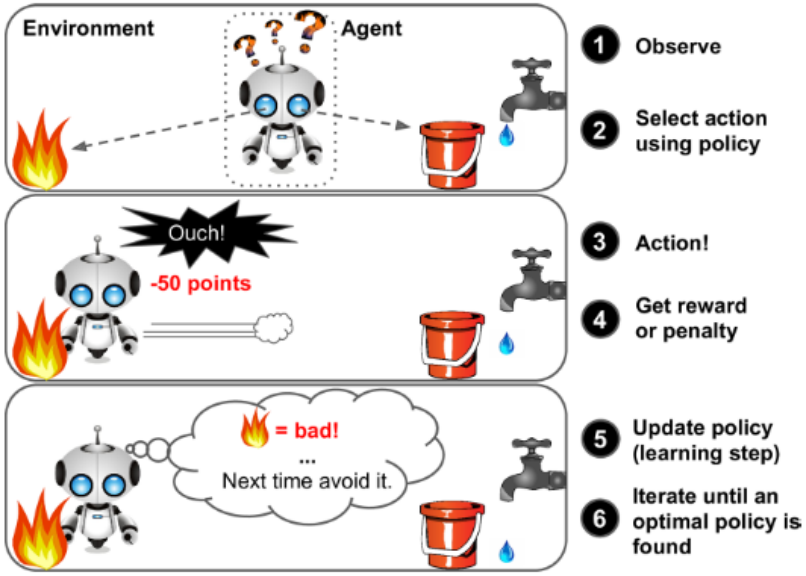


Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Batch/Online Learning

- Another criterion to classify ML systems is whether or not the system can learn incrementally from a stream of incoming data.
- In batch learning, the system must be trained using all the available data. First the system is trained, then it is delivered to apply what it has learned, and runs without learning anymore (offline learning).
- In online learning, the system is trained incrementally by feeding it data instances sequentially, individually or by small groups. Each step is fast and cheap, so the system learns about new data on the fly, as they arrive.

Batch Learning

- A batch learning system needs be trained from scratch on the full dataset in order to be up to date on new data (e.g. a new type of spam).
- Training using the full set of data can take many hours, so training would typically performed every 24 hours or weekly.
- If the system needs to adapt to rapidly changing data (e.g., to predict stock prices), then a more reactive solution is needed.
- Training on the full set of data requires a lot of computing resources. But in many contexts the system needs to learn autonomously with limited resources (e.g., a smartphone app.).
- A better option in these cases is to use online learning.

Online Learning

- Suitable for systems that receive data as a continuous flow (e.g., stock prices) and need to adapt to change rapidly or autonomously.
- Once the system has learned about new data instances, it does not need them anymore, so it can discard them (unless it wants to be able to roll back to a previous state). This can save a huge amount of space.
- Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory (out-of-core learning). The algorithm loads part of the data, runs a training step on them, and repeats the process until it has run on all of the data.
- An important parameter of online learning systems is the learning rate, which specifies how fast they should adapt to changing data: with a high learning rate the system will rapidly adapt to new data, but it will quickly forget the old ones; with a low rate it will learn more slowly, but it will be less sensitive to noise in the new data or to sequences of nonrepresentative data.

Instance vs Model Based Learning

- One more way to categorize Machine Learning systems is by how they generalize.
- The ML system needs to be able to generalize to examples it has never seen before.
- Good performance measure on the training data is insufficient, because the goal is to perform well on new instances.
- There are two main approaches to generalization: instance-based learning and model-based learning.

Instance-Based Learning

- In instance-based learning: the system learns the examples, then it generalizes to new cases using a similarity measure.
- For example, instead of flagging emails that are identical to known spam one, the spam filter could be programmed to also flag emails that are similar to known one. This requires a measure of similarity between two emails (e.g. the number of words they have in common).

Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Model-Based Learning

- In model-based learning the systems builds a model of the given examples, and uses it to make predictions.
- For example, suppose you want to know if money makes people happy, you download the Better Life Index data from OECD's website and stats about Gross Domestic Product (GDP) per capita from the International Monetary Fund (IMF)'s website.
- Then you join the tables and sort by GDP per capita:

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 from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Model-Based Learning Example

- It looks like life satisfaction goes up more or less linearly as the country's GDP per capita increases.
- So a linear model of life satisfaction with just one attribute (GDP per capita) might be suitable to model life satisfaction. This step is called model selection: $\text{life\_satisfaction} = \theta_0 + \theta_1 \times \text{GDP\_per\_capita}$
- By tweaking θ_0 and θ_1 parameters, the model can represent any linear function.

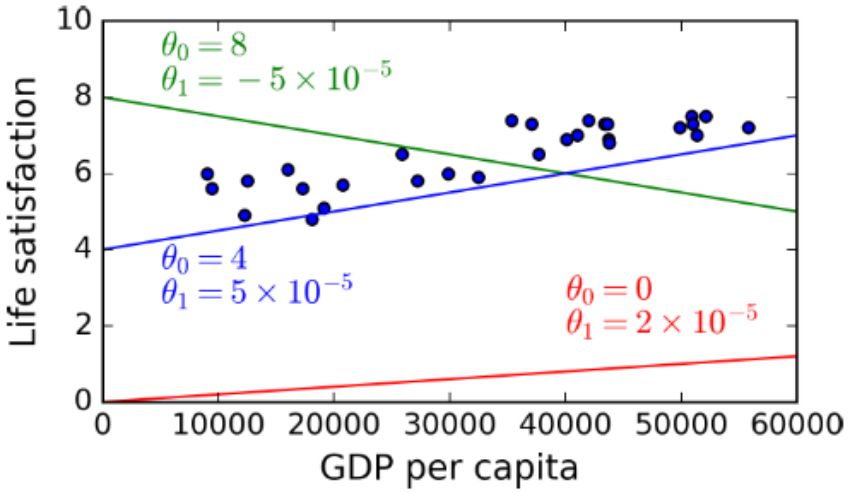


Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Model-Based Learning Example(2)

- To use the model it is necessary to define θ_0 and θ_1 so as to make the model perform best.
- A utility function (or fitness function) can be used to measure how good the model is, or equivalently, a cost function can measure how bad it is.
- For linear regression problems, it is typically used a cost function measuring the distance between the linear model's predictions and the training examples, aiming to minimize this distance.
- The Linear Regression algorithm can be used to find the parameters that make the linear model fit best to input data, starting from training examples. This is called training the model.

Model-Based Learning Example(3)

- In our example the algorithm finds that the optimal parameter values are:

$$\theta_0 = 4.85 \text{ and } \theta_1 = 4.91 \times 10^{-5}$$

- Suppose we want to run the model to make predictions, say for Cypriots, whose GDP per capita is $ 22,587, but for which the OECD data does not provide the answer.
- We can apply the learned model and find that life satisfaction is likely to be somewhere around:

$$4.85 + 22{,}587 \times 4.91 \times 10^{-5} = 5.96$$

Model-Based Learning Example(4)

- Using instance-based learning algorithm instead, we would have found that Slovenia has the closest GDP per capita to that of Cyprus ($20,732), and since the OECD data tells us that Slovenians' life satisfaction is 5.7, we would have predicted a life satisfaction of 5.7 for Cyprus.
- If we look at the two next closest countries, Portugal and Spain, with life satisfactions of 5.1 and 6.5, respectively, averaging these three values we get 5.77, which closer to the model-based prediction.
- This simple algorithm is called k-Nearest Neighbors regression (in this example, k = 3).

Modeling Alternatives

- If the selected model does not guarantee good prediction performances, we can either use more attributes (employment rate, health, air pollution, etc.), get more or better quality training data, or select a more powerful model (e.g., a Polynomial Regression model).

Model Based Learning Summary

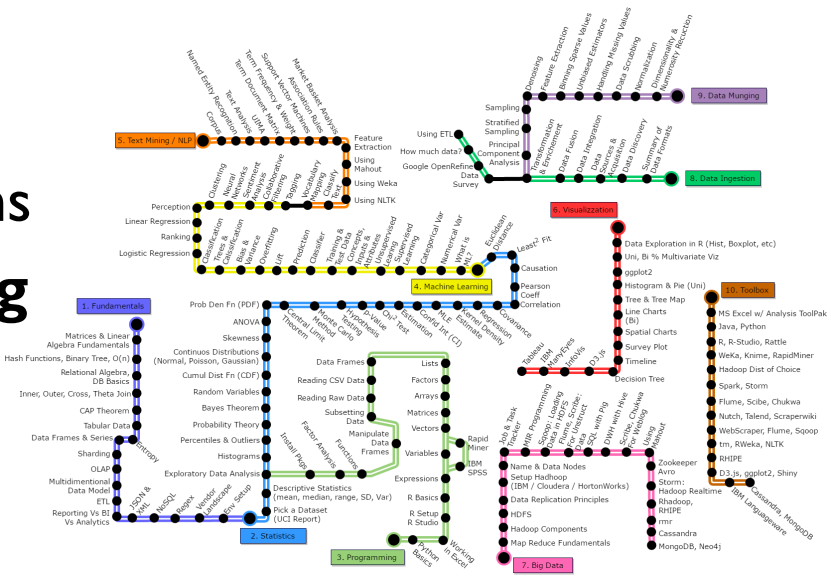
- In summary:
  - We studied the data;
  - We selected a model;
  - We trained it on the training data (i.e., the learning algorithm searched for the model parameter values minimizing a cost function).
  - Finally, we applied the model to make predictions on new cases (inference), hoping that the model will generalize well.
- This is what a typical Machine Learning project looks like.

Frameworks for Machine Learning

- Rather than implementing ML algorithms, production-ready frameworks can be used, like these based on Python:
  - Scikit-Learn implements many Machine Learning algorithms efficiently.
  - TensorFlow is a library for distributed numerical computation using data flow graphs. It makes it possible to train and run large neural networks efficiently by distributing the computations across potentially thousands of multi-GPU servers. TensorFlow was created at Google and supports many of their large-scale Machine Learning applications. It was open-sourced in November 2015.

Overview

1. Some History
2. Machine Learning Examples
3. Goals of Machine Learning
4. Types of Machine Learning Systems
5. **Problems in Machine Learning**
6. Application scenarios
7. Summary



Problems in Machine Learning

- The following are among the most relevant problems in Machine Learning:
  - Insufficient Quantity of Training Data;
  - Nonrepresentative Training Data;
  - Poor-Quality Data;
  - Irrelevant Features;
  - Overfitting the Training Data;
  - Underfitting the Training Data;

Insufficient Quantity of Training Data

- It takes a lot of data for ML algorithms to work properly.
- Even for very simple problems thousands of examples are needed, and for complex problems such as image or speech recognition millions of examples are needed.
- In a 2001 paper Microsoft researchers Michele Banko and Eric Brill showed that very different ML algorithms performed almost identically well on a complex problem of natural language disambiguation, once they were given enough data.

Nonrepresentative Training Data

- It is crucial that training data be representative of the new cases we want to generalize to, whether we use instance-based or model-based learning.
- For example, the countries we used earlier for training the linear model were not perfectly representative.

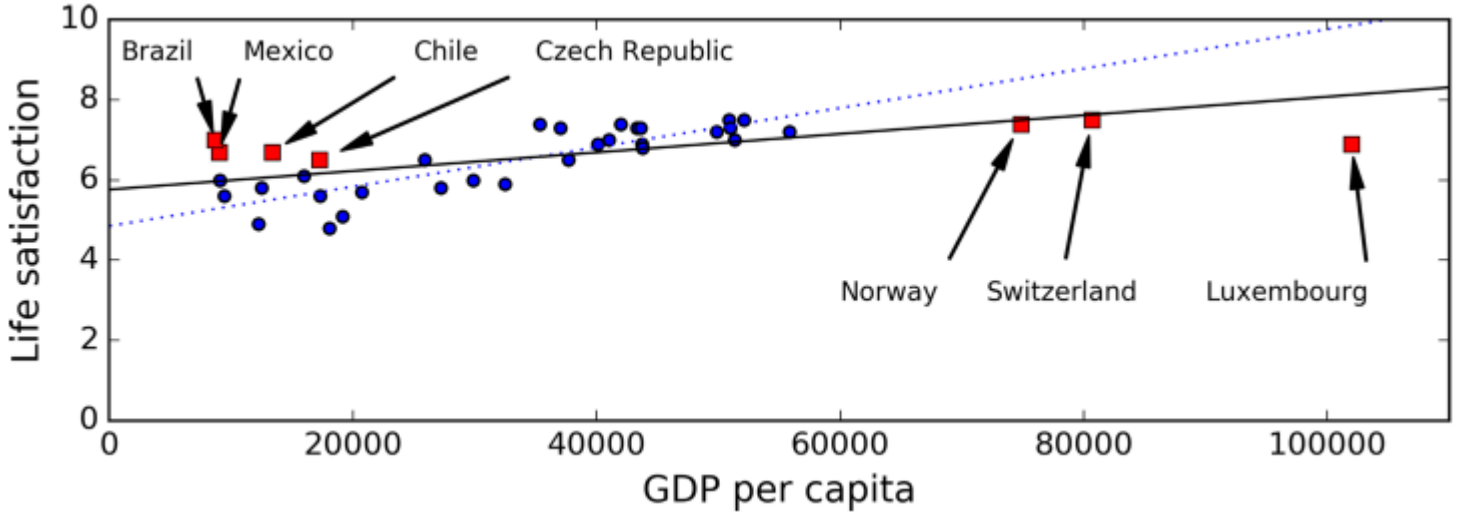


Figure from *Aurelien Geron: «Hands on Machine Learning with Scikit Learn and TensorFlow, O'Reilly ed.*

Nonrepresentative Training Data(2)

- Training a linear model also on this data yields the solid line, while the old model is represented by the dotted line.
- Thus, adding a few missing countries significantly alter the model, and it makes it clear that a simple linear model is probably never going to work well. It seems that very rich countries are not happier than moderately rich countries, and conversely some poor countries seem happier than many rich countries.
- By using a nonrepresentative training set, we trained a model that is unlikely to make accurate predictions, especially for very poor and very rich countries.

Poor-Quality Data

- If training data is full of errors, outliers, and noise (e.g., due to poor quality measurements), it will make it harder for the system to detect the underlying patterns, and to perform well.
- It is worth the effort to spend time cleaning up training data.
- Most data Scientists spend a significant part of their time doing just that.
- For example, if some instances are outliers, it may help to simply discard them or try to fix the errors manually.
- If some instances are missing a few features (e.g., 5% of customers did not specify their age), we must decide whether to ignore this attribute, ignore these instances, fill in the missing values (e.g., with the median age), and so on.

Irrelevant Features

- Garbage in, garbage out: the system will only be capable of learning if the training data contains enough relevant features and not too many irrelevant ones.
- A critical part of the success of a ML project is coming up with a good set of features to train on. This process, called feature engineering, involves:
  - Feature selection: selecting the most useful features to train on among existing features.
  - Feature extraction: combining existing features to produce a more useful one (dimensionality reduction).
  - Creating new features by gathering new data.

Overfitting the Training Data

- Overfitting means that the model performs well on the training data, but it does not generalize well.
- For example, imagine to feed the life satisfaction model many more attributes, including the uninformative country's name.
- A complex model may detect patterns like: *all countries in the training data with a w in their name have a life satisfaction greater than 7*: Norway (7.4), Sweden (7.2), and Switzerland (7.5).
- How confident we are that the W-satisfaction rule generalizes to Rwanda or Zimbawe?
- Obviously, this pattern occurred in the training data by pure chance, but the model has no way to tell whether a pattern is real or simply the result of noise in the data.

Reduce the Risk of Overfitting

- Overfitting happens when the model is too complex w.r.t. the amount and noisiness of the training data.
- Possible solutions are:
  - To select a model with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data;
  - To gather more training data;
  - To reduce the noise in the training data (e.g., fix data errors and remove outliers);
- Constraining a model to make it simpler and reduce the risk of overfitting is called regularization.

Underfitting the Training Data

- Underfitting occurs when the model is too simple to learn the underlying structure of the data.
- For example, a linear model of life satisfaction is prone to underfit; reality is more complex than the model, so its predictions are bound to be inaccurate, even on the training examples.
- The main options to fix this problem are:
  - Selecting a more powerful model, with more parameters;
  - Feeding better features to the learning algorithm;
  - Reducing the constraints on the model.

Testing and Validation

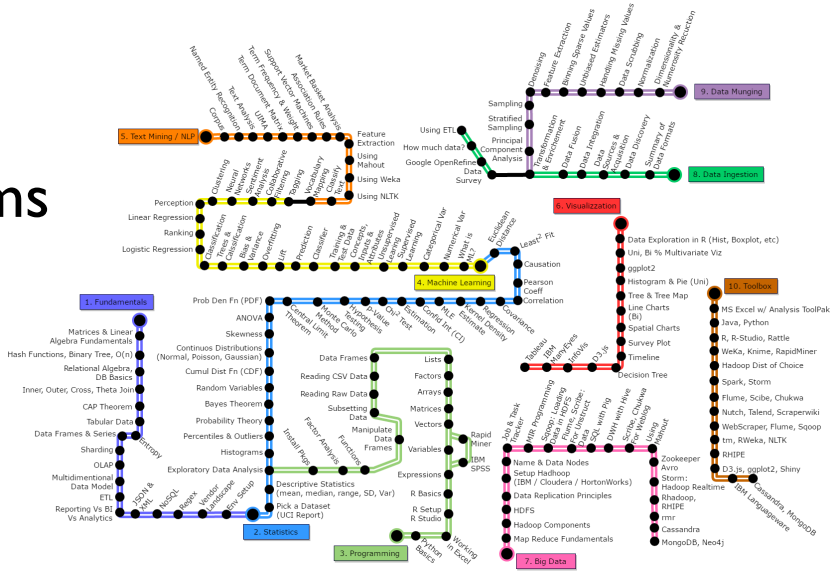
- One way to do validate a ML model is to put it in production and monitor how well it performs.
- But if the model is bad, users will complain !
- A better option is to split your data into two sets: the training set and the test set.
- It is common to use 80% of the data for training and 20% for testing.
- If the model makes few mistakes on the training set, and many on the test set it means that the model is overfitting the training data.

Cross-validation

- To avoid "wasting" training data in validation sets, a common technique is to use cross-validation: the training set is split into complementary subsets, and each model is trained against a different combination of these subsets and validated against the remaining parts.
- Once the model type and hyperparameters have been selected, a final model is trained using these hyperparameters on the full training set, and the generalized error is measured on the test set.

Overview

1. Some History
2. Machine Learning Examples
3. Goals of Machine Learning
4. Types of Machine Learning Systems
5. Problems in Machine Learning
6. **Application scenarios**
7. Summary



Predicting Customer Churn

You just landed a great analytical job with MegaTelCo, one of the largest telecommunication firms in the US.

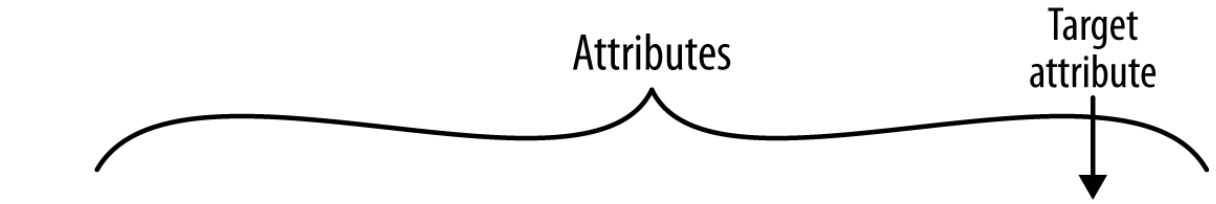
They are having a major problem with customer retention in their wireless business.

- In the mid-Atlantic region, 20% of cell phone customers leave when their contracts expire. Communications companies are now engaged in battles to attract each other's customers while retaining their own.
- Marketing has already designed a special retention offer.
- Your task is to devise a precise, step-by-step plan for how the data science team should use MegaTelCo's vast data resources to solve the problem.

Predicting Customer Churn (2)

- What data you might use?
- How would they be used?
- How should MegaTelCo choose a set of customers to receive their offer in order to best reduce churn for a particular incentive budget?

Terminology



Name	Balance	Age	Employed	Write-off
Mike	$200,000	42	no	yes
Mary	$35,000	33	yes	no
Claudio	$115,000	40	no	no
Robert	$29,000	23	yes	yes
Dora	$72,000	31	no	no

This is one row (example).
Feature vector is: **<Claudio,115000,40,no>**
Class label (value of Target attribute) is **no**

Common ML Tasks

- Classification and regression
  - How likely is this consumer to respond to our campaign?
- Regression
  - How much will she use the service?
- Similarity Matching
  - Can we find consumers similar to my best customers?
- Clustering
  - Do my customers form natural groups?
- Causal Modeling
  - Why are my customers leaving?

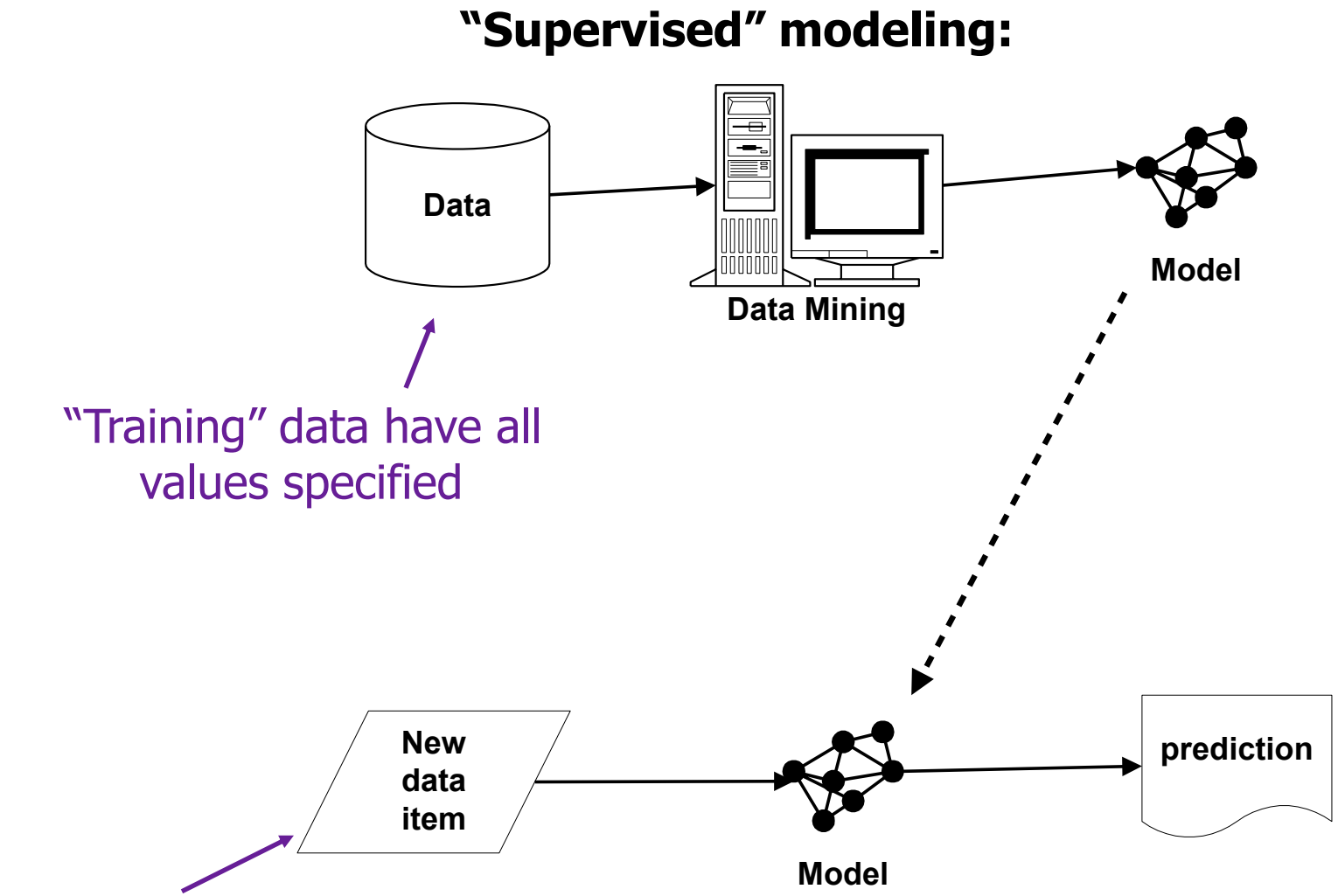
Supervised vs Unsupervised

- "Do our customers naturally fall into different groups?"
  - No guarantee that the results are meaningful or will be useful for any particular purpose
- "Can we find groups of customers who have particularly high likelihoods of canceling their service soon after contracts expire?"
  - **A specific purpose**
  - Much more useful results (usually)
  - Different techniques
  - **Requires data on the target**
    - The individual's label

Subclasses of Supervised ML

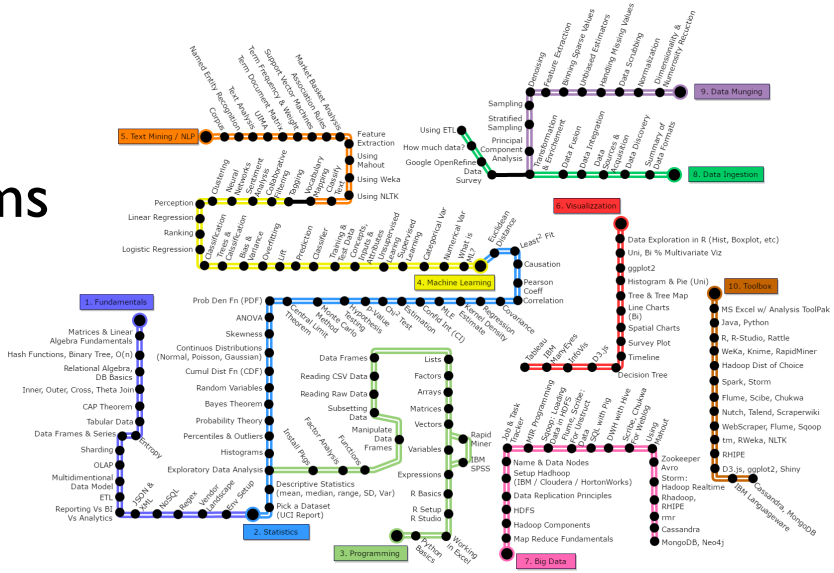
- "Will this customer purchase service $S1$ if given incentive $I1$?"
  - Classification problem
    - Binary target (the customer either purchases or does not)
- "Which service package ($S1$, $S2$, or none) will a customer likely purchase if given incentive $I1$?"
  - Classification problem
    - Three-valued target
- "How much will this customer use the service?"
  - Regression problem
    - Numeric target
    - Target variable: amount of usage per customer

ML versus Use of the Model



Overview

1. Some History
2. Machine Learning Examples
3. Goals of Machine Learning
4. Types of Machine Learning Systems
5. Problems in Machine Learning
6. Application scenarios
7. **Summary**



Summary

- Machine Learning is about learning from data, instead of having to explicitly code rules.
- In a ML project data are gathered in a training set, and we feed the training set to a learning algorithm.
- The system will not perform well if the training set is too small, or if the data is not representative, noisy, or polluted with irrelevant features. Lastly, the model needs to be neither too simple (in which case it will underfit) nor too complex (in which case it will overfit).
- Once a model is trained, it needs to be evaluated and finetuned if necessary.