Scikit-Learn

INTRODUCTION

Scikit-Learn is very easy to use, yet it implements many Machine Learning algorithms efficiently, so it makes for a great entry point to learning Machine Learning. It was created by David Cournapeau in 2007, and is now led by a team of researchers at the French Institute for Research in Computer Science and Automation (Inria).

Tensorflow

INTRODUCTION

TensorFlow is a more complex library for distributed numerical computation. It makes it possible to train and run very large neural networks efficiently by distributing the computations across potentially hundreds of multi-GPU (graphics processing unit) servers. TensorFlow (TF) was created at Google and supports many of its large-scale Machine Learning applications. It was open sourced in November 2015, and version 2.0 was released in September 2019.

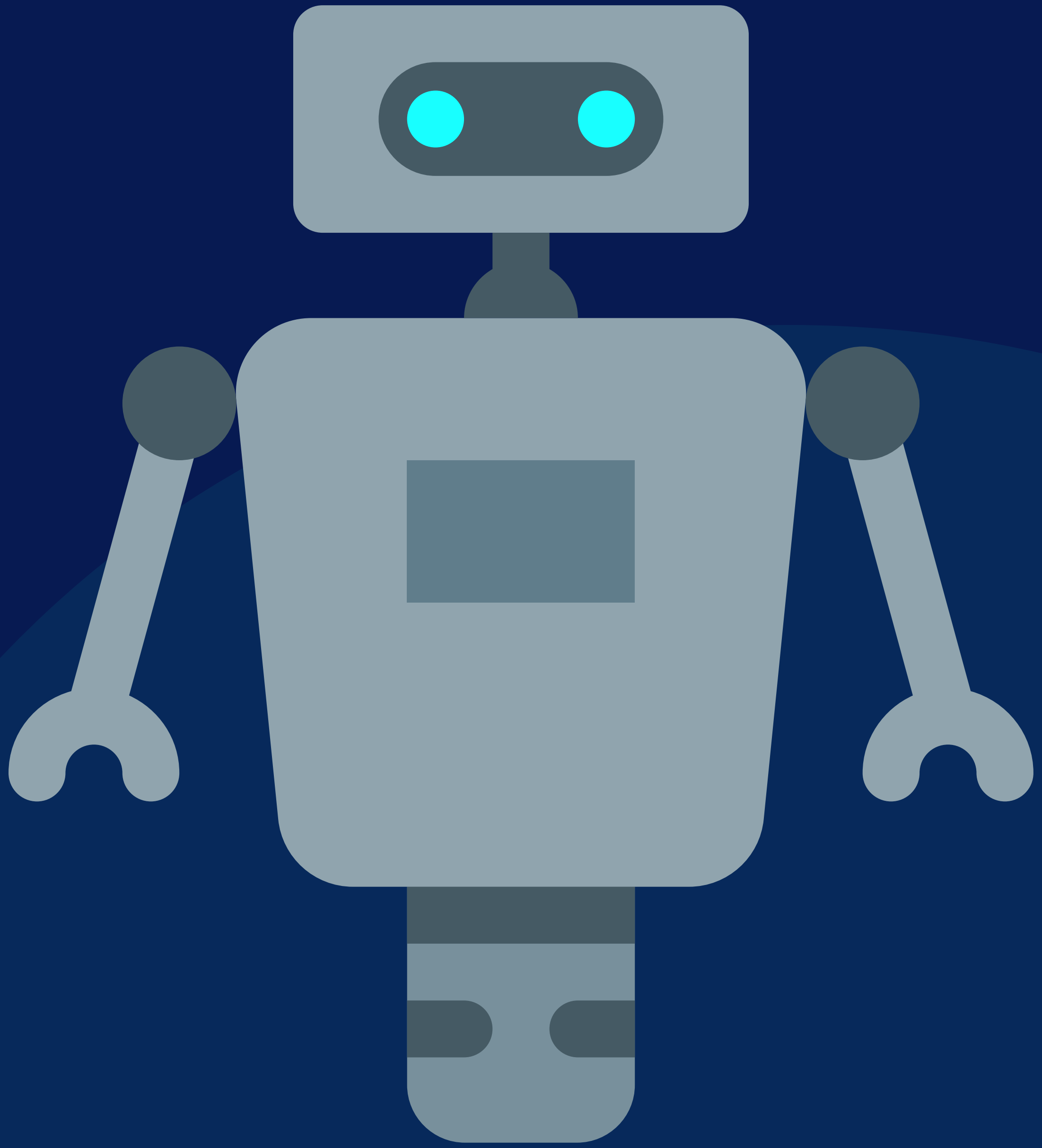
Keras

INTRODUCTION

Keras is a high-level Deep Learning API that makes it very simple to train and run neural networks. It can run on top of either TensorFlow, Theano, or Microsoft Cognitive Toolkit (formerly known as CNTK). TensorFlow comes with its own implementation of this API, called tf.keras, which provides support for some advanced TensorFlow features (e.g., the ability to efficiently load data).

MACHINE LEARNING FOR COMPLETE BEGINNERS

(THE BEST TRAINING PROGRAM ON PLANET)



SHAPEAI

LET'S START WITH WHAT IS MACHINE LEARNING ?

(AN INTRODUCTION TO MACHINE LEARNING)

SHAPEAI

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- Here is a slightly more general definition:

> [Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.
>
> —Arthur Samuel, 1959

SHAPEAI

SPAM MAIL CLASSIFICATION USING MACHINE LEARNING ?

(SPAM OR HAM CLASSIFICATION)

SHAPEAI

Consider how you would write a spam filter using traditional programming techniques:

A spam filter based on Machine Learning techniques :

A spam filter based on Machine Learning techniques that automatically adapts to changes :

Finally, Machine Learning can help humans learn :

To summarize, Machine Learning is great for:

- Problems for which existing solutions require a lot of fine-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better than the traditional approach.
- Complex problems for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution.
- Fluctuating environments: a Machine Learning system can adapt to new data.
- Getting insights about complex problems and large amounts of data.

APPLICATIONS OF MACHINE LEARNING ?

(HOW CAN WE USE MACHINE LEARNING)

SHAPEAI

- Analyzing images of products on a production line to automatically classify them:
- Detecting tumors in brain scans
- Automatically classifying news articles
- Automatically flagging offensive comments on discussion forums
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Forecasting your company’s revenue next year
- Making your app react to voice commands
- Detecting credit card fraud
- Segmenting clients based on their purchases so that you can design a different marketing strategy for each segment
- Recommending a product that a client may be interested in, based on past purchases

TYPES OF MACHINE LEARNING SYSTEMS ?

(BROAD CATEGORIES OF MACHINE LEARNING)

SHAPEAI

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories, based on the following criteria:

- Whether or not they are trained with human supervision (supervised, unsupervised, semisupervised, and Reinforcement Learning).
- Whether or not they can learn incrementally on the fly (online versus batch learning).
- Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (instance-based versus model-based learning)

SUPERVISED/ UNSUPERVISED LEARNING

(TYPE OF SUPERVISION THEY GET DURING TRAINING)

SHAPEAI

There are four major categories:

- Supervised Learning
- Unsupervised Learning
- Semisupervised Learning, and
- Reinforcement Learning.

In this part we will go through all the types of learning trying to get a high level intution and understanding behind all the learning techniques

Supervised learning

In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels. We can the data being books which we use to teach the model(students).

Here are some of the most important supervised learning algorithms (covered in this course:

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

Unsupervised learning

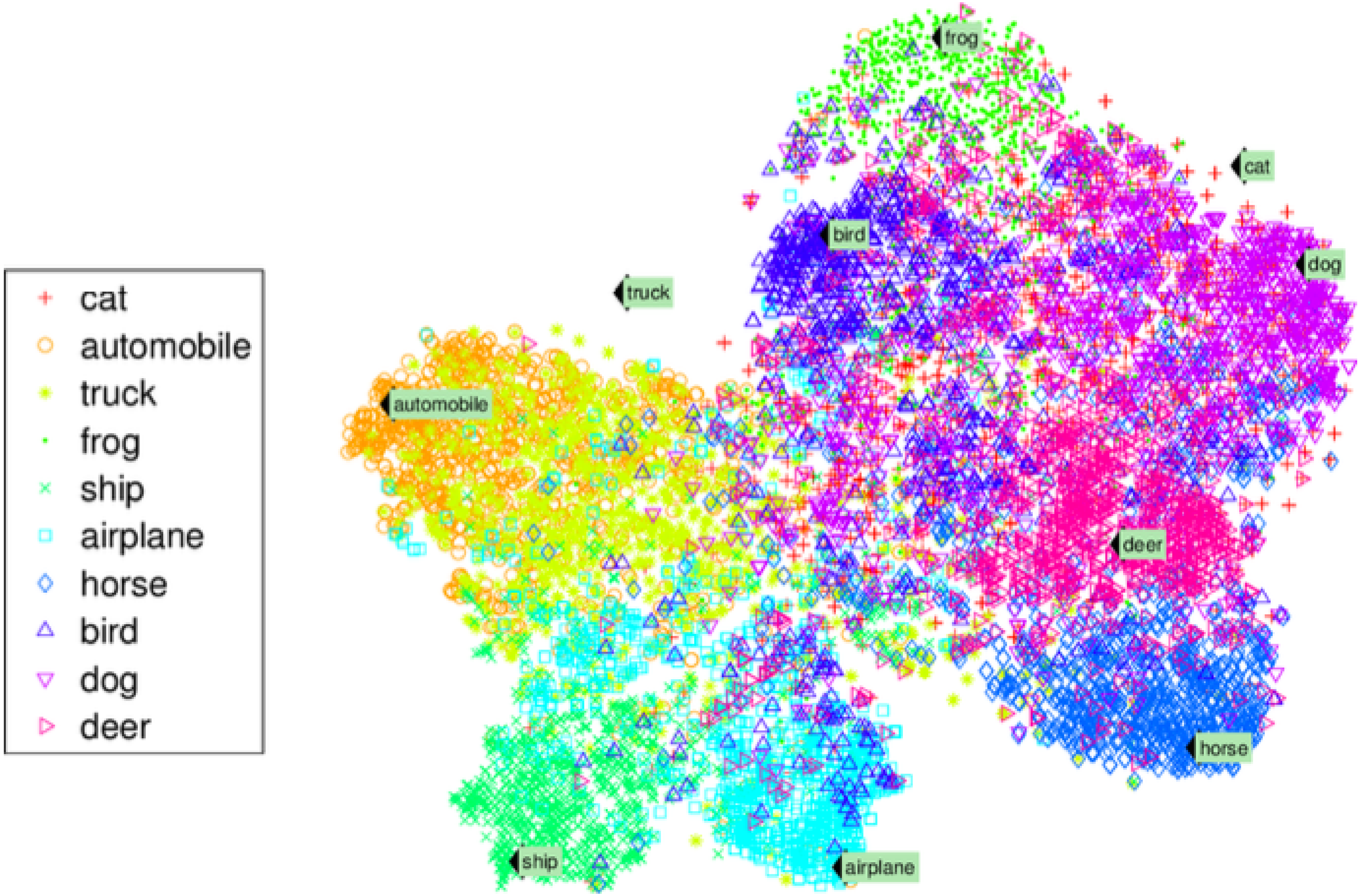
In unsupervised learning, as you might guess, the training data is unlabeled. The system tries to learn without a teacher.

Here are some of the most important unsupervised learning algorithms (covered in this course):

- Clustering
  — K-Means
  — DBSCAN
  — Hierarchical Cluster Analysis (HCA)
- Anomaly detection and novelty detection
  — One-class SVM
  — Isolation Forest
- 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

Clustering Algorithm

Visualization Algorithm



Example of a t-SNE visualization highlighting semantic clusters

Dimensionality Reduction

The goal is to simplify the data without losing too much information

Anaomaly & Novelty Detection

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.

Association Rule Learning

The goal is to dig into large amounts of data and discover interesting relations betweendiffrent attributes.

Semisupervised learning

Since labeling data is usually time-consuming and costly, you will often have plenty of unlabeled instances, and few labeled instances. Some algorithms can deal with data that's partially labeled. This is called semisupervised learning

Reinforcement learning

Reinforcement Learning is a very different beast. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return (or penalties in the form of negative rewards). It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

BATCH / ONLINE LEARNING

(LEARNING INCREMENTALLY FROM A STREAM OF INCOMING DATA)

SHAPEAI

Batch learning

In batch learning, the system is incapable of learning incrementally: it must be trained using all the available data. This will generally take a lot of time and computing resources, so it is typically done offline. First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned. This is called offline learning.

If your system needs to be able to learn autonomously and it has limited resources (e.g., a smartphone application or a rover on Mars), then carrying around large amounts of training data and taking up a lot of resources to train for hours every day is a showstopper.Fortunately, a better option in all these cases is to use algorithms that are capable of learning incrementally.

Online learning

In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or in small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives.

Online learning is great for systems that receive data as a continuous flow (e.g., stock prices) and need to adapt to change rapidly or autonomously. It is also a good option if you have limited computing resources: once an online learning system has learned about new data instances, it does not need them anymore, so you can discard them (unless you want to be able to roll back to a previous state and "replay" the data). 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 (this is called out-of-core learning). The algorithm loads part of the data, runs a training step on that data, and repeats the process until it has run on all of the data.

Out-of-core learning is usually done offline (i.e., not on the live system), so online learning can be a confusing name. Think of it as incremental learning.

Now a days it is increasingly being replaced by distributed learning as in spark, pig, hive etc. This comes in big data.

INSTANCE-BASED / MODEL-BASED LEARNING

(HOW WELL DO THEY GENERALIZE)

SHAPEAI

Instance-Based learning

Instance-based learning: the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples (or a subset of them).

Model-Based learning

Another way to generalize from a set of examples is to build a model of these examples and then use that model to make predictions. This is called model-based learning

In summary:

- You studied the data.
- You selected a model.
- You trained it on the training data (i.e., the learning algorithm searched for the model parameter values that minimize a cost function).
- Finally, you applied the model to make predictions on new cases (this is called inference), hoping that this model will generalize well.

We have covered a lot of ground so far: you now know what Machine Learning is really about, why it is useful, what some of the most common categories of ML sys- tems are, and what a typical project workflow looks like.

MAIN CHALLENGES OF MACHINE LEARNING

(“BAD ALGORITHM” AND “BAD DATA”)

SHAPEAI

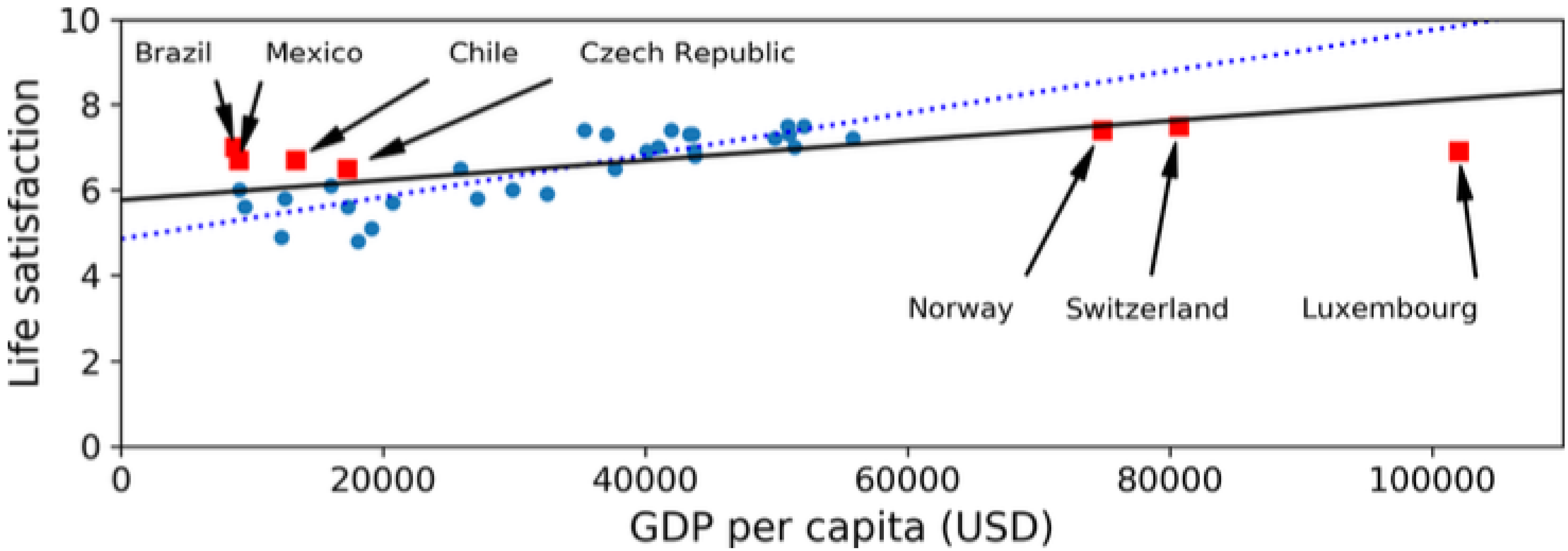
Insufficient Training Data

It takes a lot of data for most Machine Learning algorithms to work properly. Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples (unless you can reuse parts of an existing model called as transfer learning which will be discussed later).

Nonrepresentative Training Data

In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to. This is true whether you use instance-based learning or model-based learning.

For example, the set of countries we used earlier for training the linear model was not perfectly representative; a few countries were missing.


Brazil
Mexico
Chile
Czech Republic
Norway
Switzerland
Luxembourg
Life satisfaction
0
2
4
6
8
10
0
20000
40000
60000
80000
100000
GDP per capita (USD)

SHAPEAI

It is crucial to use a training set that is representative of the cases you want to general- ize to. This is often harder than it sounds: if the sample is too small, you will have sampling noise (i.e., nonrepresentative data as a result of chance), but even very large samples can be nonrepresentative if the sampling method is flawed. This is called sampling bias.

Poor-Quality Data

The following are a couple of examples of when you’d want to clean up training data:

- If some instances are clearly 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 your customers did not specify their age), you must decide whether you want to ignore this attribute altogether, ignore these instances, fill in the missing values (e.g., with the median age), or train one model with the feature and one model without it.

SHAPEAI

Irrelevant Features

The following are a couple of examples of when you’d want to clean up training data:

- If some instances are clearly 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 your customers did not specify their age), you must decide whether you want to ignore this attribute altogether, ignore these instances, fill in the missing values (e.g., with the median age), or train one model with the feature and one model without it.

Overfitting

Overfitting refers to a model that models the training data too well. Overfittinghappens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.

Complex models such as deep neural networks can detect subtle patterns in the data, but if the training set is:

- Noisy
- Small
- Introduced sampling noise,

Then the model is likely to detect patterns in the noise itself.

Overfitting happens when the model is too complex relative to the amount and noisiness of the training data. Here are possible solutions:

- Simplify the model by selecting one 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, or by constraining the model.
- Gather more training data.
- 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.

END-TO-END MACHINE LEARNING PROJECT

(WELCOME TO YOUR FIRST JOB AS DATA SCIENTIST)

SHAPEAI

Overfitting happens when the model is too complex relative to the amount and noisiness of the training data. Here are possible solutions:

- Simplify the model by selecting one 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, or by constraining the model.
- Gather more training data.
- 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.