

DevCon 4

Machine Learning Chapter

By Faris Hassan

Trainers

Faris Hassan (Farisology)

What is artificial
Intelligence?

How Does a
machine learn to
recognize objects?

How do we
humans learn?

How did you
learn to recognize
blue color?

How did you
recognize
cars/dogs/cats?

How Does a
machine learn to
recognize objects?

Preface

—

Machine Learning

History

- Definition
- Types
- Evolution
- Steps

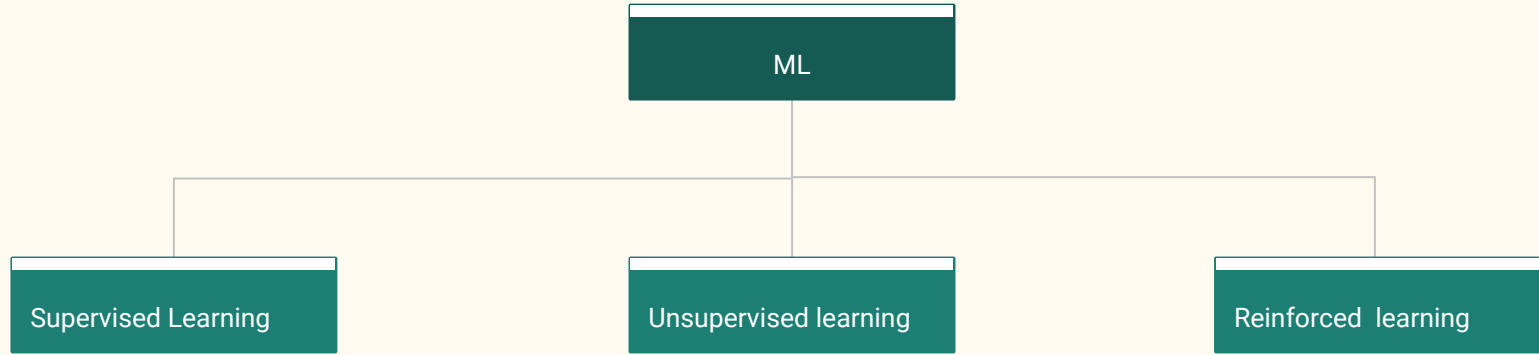


Definition

Arthur 1959: the subfield of computer science that gives computers the ability to learn without being explicitly programmed.

Mitchel 1997: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .

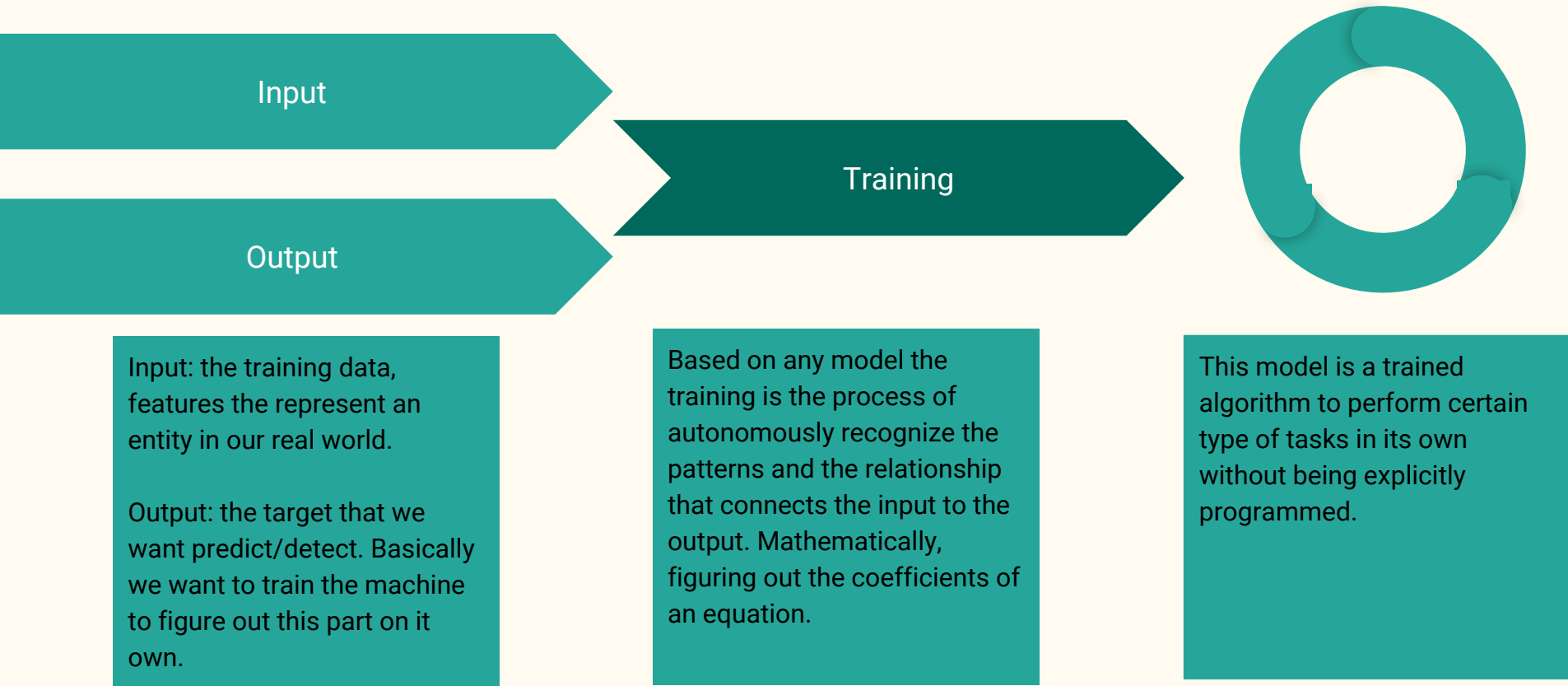
Types of machine learning



Difference between ML and Sequential Logic



Machine learning Logic



Supervised Learning

Scope

- Classification
- Regression



Classification

Predicting Categories

Types of an entity

Bad - Good - Medium

Sick - not sick

Hot dog- not hot dog

Sad - happy - surprised - angry

Regression

Predicting values

Continuous values

- Sales
- Coordinates
- Time
- Age
- Power
- pressure

Python stack for ML

—

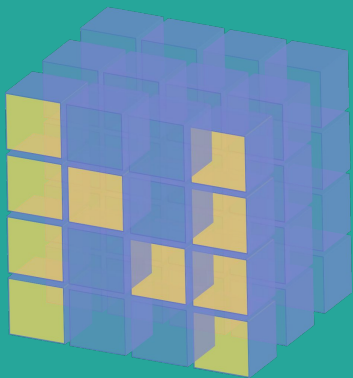
What tools do we
use to practice
ML using python
language?

Python Stack

Machine Learning

- Numpy
- Pandas
- Opencv (for images)
- NLTK (for languages)
- Scikit learn (for machine learning)
- Tensorflow + keras (deep learning)





NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

Create a numpy list

```
Import numpy as np
```

```
A = [1,2,3,4,5,6]
```

```
A1 = np.array(a, dtype=np.int16)
```

```
Import numpy as np
```

```
A = [[1,2,3], [4,5,6]]
```

```
A1 = np.array(a, dtype=np.float32)
```

Numpy

Attributes:

`a.ndim` # dimension of array p

`a.shape` # size of array dimension

`len(a)` # length of array

`a.dtype` # data type of array

Supported data types:

`bool`

`int8, uint8, int16, uint16, int64, uint64`

`float16, float32, float64`

`complex64, complex128, complex256`

`string`

Arithmetic

```
a = [1,2,3,4]
```

```
b = [4,3,2,1]
```

```
a1 = np.array(a)
```

```
b1 = np.array(b)
```

```
a1+b1 # different from Python list
```

```
a1-b1 # can't do using Python list
```

```
a1*b1
```

```
a1/b1
```

What do you observe ?

Do you see the arithmetic power of numpy?

Try this

```
a = np.arange(2,20,3)
```

```
a = np.linspace(2,20,3)
```

```
np.random.seed(25)
```

```
a = np.random.randn(1000)*4+3
```

```
a = np.exp(2)
```

```
a = np.sqrt(4)
```

```
a = np.sin(np.pi/2)
```

```
a = np.cos(np.pi/2)
```


Statistical numpy attributes

```
a = np.random.randn(100)*2+5
```

```
np.mean(a)
```

```
np.mean(a,axis=1)
```

```
np.var(a)
```

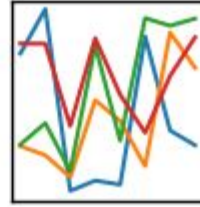
```
np.std(a)
```

How easy it is to calculate your statistical measure in numpy?

Such a darling!!

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Flexible and powerful data analysis / manipulation library for Python, providing high-performance, easy-to-use data structures, data.frame objects, statistical functions, and data analysis tools for the [Python](#) programming language.

Main structures in Pandas

Dataframes

- DataFrame is similar to a spreadsheet in Excel
- It has rows index and columns name

Series

- Series is the primary building block of pandas
- It represents a one-dimensional labeled Numpy array

Creating series

```
a = np.random.randn(100)*5+100
```

```
date = pd.date_range('20170101',periods=100)
```

```
s = pd.Series(a,index=date)
```

```
import pandas as pd
```

```
pd.Series([1,3,5,6])
```

```
pd.Series([1,3,5,6], index=['A1','A2','A3','A4'])
```

Creating dataframes

```
d = [[1,2],[3,4]]
```

```
df =  
pd.DataFrame(d1,index=[1,2],columns=['a','b'])
```

```
d = np.arange(24).reshape(6,4)
```

```
df = pd.DataFrame(d, index=np.arange(1,7),  
columns=list('ABCD'))
```

Dataframe/Series attributes

`df.shape` : Dimensionality of a DF

`df.columns` : columns of a DF

`df.index` : index of a DF

`df.values` : values of a DF

`s.index` : show the indexes

`s.values` : show the values

`len(s)` : number of elements

You can get data straight from the web

```
pip install pandas-datareader
```

```
from pandas_datareader import data,wb
```

```
msft = data.DataReader("MSFT", "yahoo",  
"2018-1-1", "2018-1-11")
```

```
pip install quandl
```

```
conda install quandl
```

```
import quandl
```

```
msft =  
quandl.get("YAHOO/MSFT",trim_start='2017-01-01'  
,trim_stop='2017-01-30')
```

```
msft.tail()
```

The experiment

