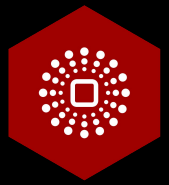
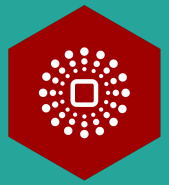
OOPs and Numpy



COGNIBOT

AI meets Industry

Object oriented programming

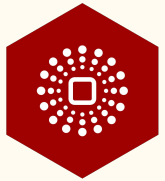


COGNIBOT

AI meets Industry

Basic foundations

- Python is a multi-paradigm language (supports different programming approaches)
- Objects contain both attributes and behaviour. (eg. a Human object would contain the attributes name, gender, height and would contain the behaviours including run, walk, sing dance etc.)

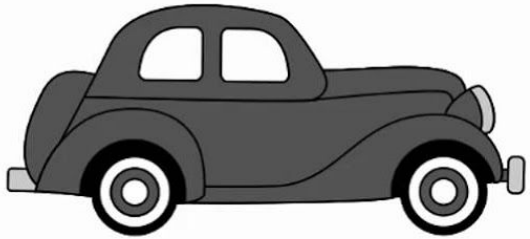


COGNIBOT

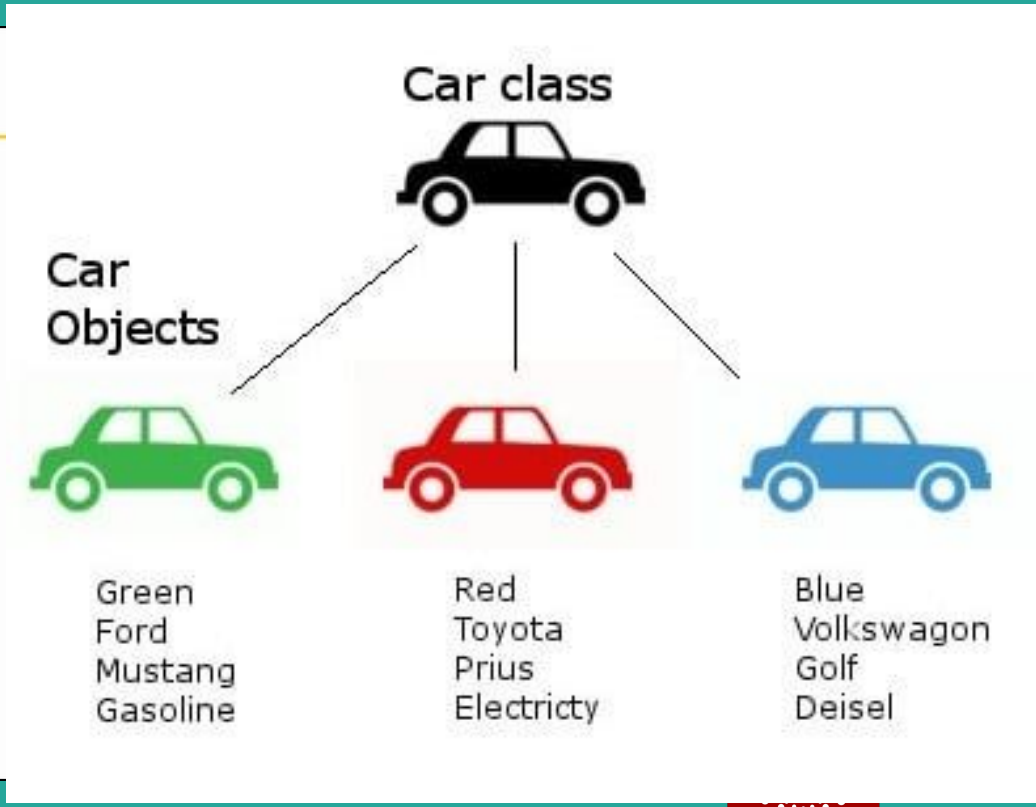
AI meets Industry

OBJECT-ORIENTED PROGRAMMING

Properties
Make
Model
Color
Year
Price

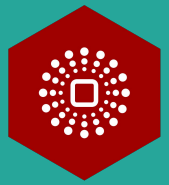


Methods
Start
Drive
Park



COGNIBOT
AI meets Industry

See Jupyter Notebook

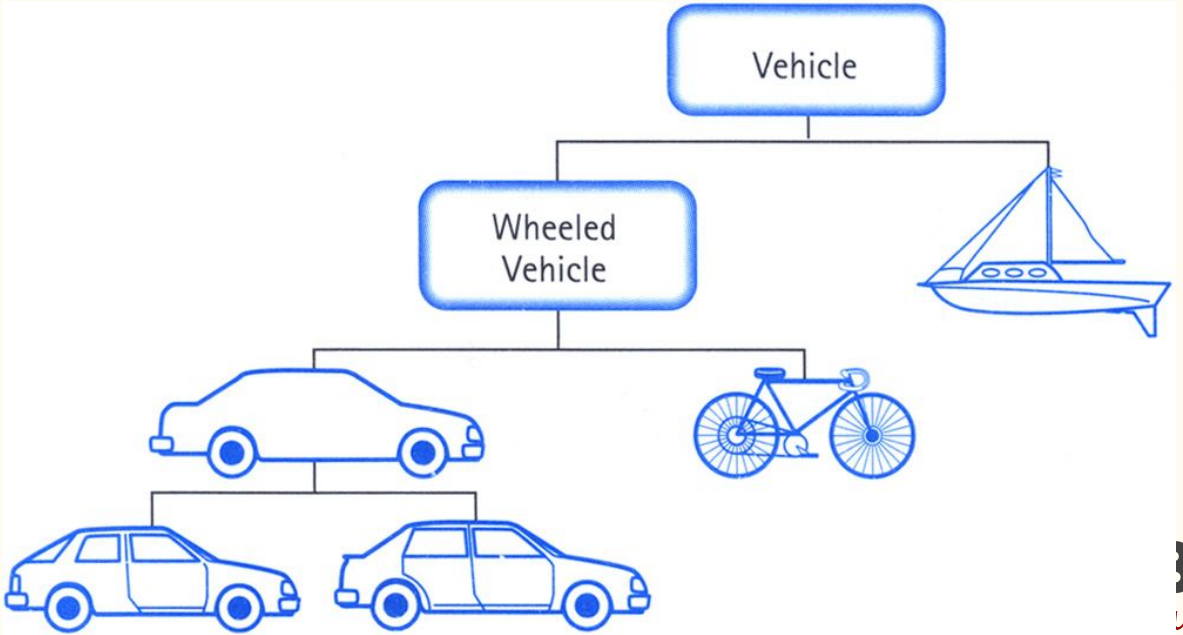


COGNIBOT

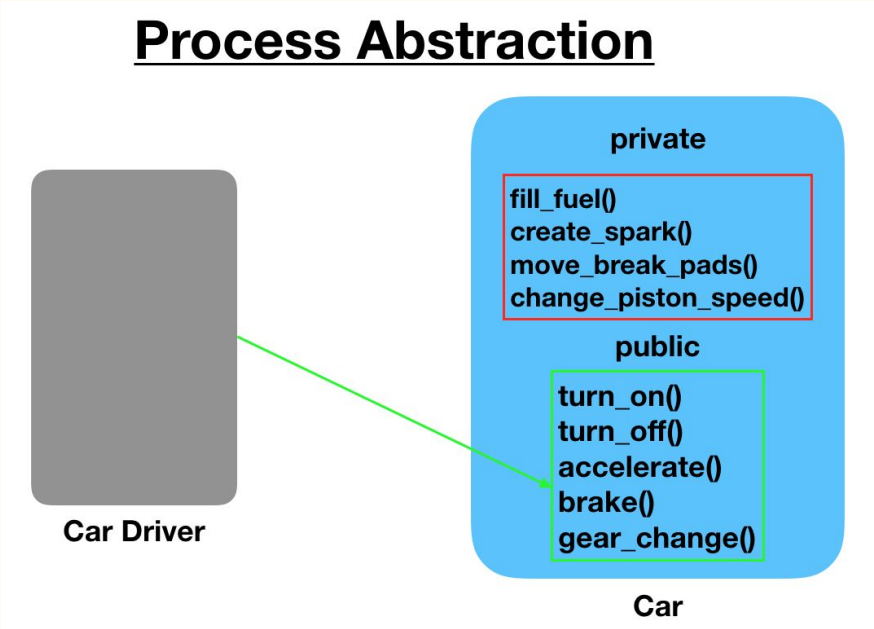
AI meets Industry

Basic OOP principles

- Encapsulation - data inside the object should only be accessed through the object's methods.
- Inheritance



- Abstraction - providing only the essential information to the user and hiding the actual implementation
- Polymorphism - A concept of using common operation in different ways for different data input. (ex print() function can print any data type)



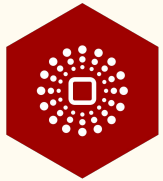
Procedural vs OOPs

Procedural

- Step-by-step approach
- Program is a list of procedures to run (functions, global variable etc.)

OOPs

- Uses classes and objects to create models
- Objects (instances of classes) contain methods and data

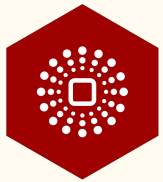


COGNIBOT

AI meets Industry

Advantages of OOPs

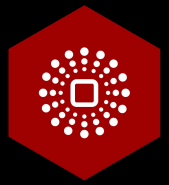
- Easier to maintain large codebase
- Better models the real world
- Enables code reuse
- Better than modules because the data is also joined with



COGNIBOT

AI meets Industry

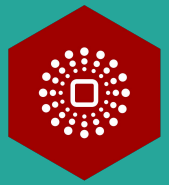
Python Classes



COGNIBOT

AI meets Industry

See Jupyter Notebook



COGNIBOT

AI meets Industry

Numpy

COGNIBOT

AI meets Industry

What is NumPy?

- A python module that adds support for multi-dimensional arrays
- Much much faster than native python
- It is implemented in C and Fortran so when calculations are vectorized (formulated with vectors and matrices), performance is very good.
- NumPy arrays are like Python's built-in list type, but NumPy arrays provide much more efficient storage and data operations as the arrays grow larger in size

COGNIBOT

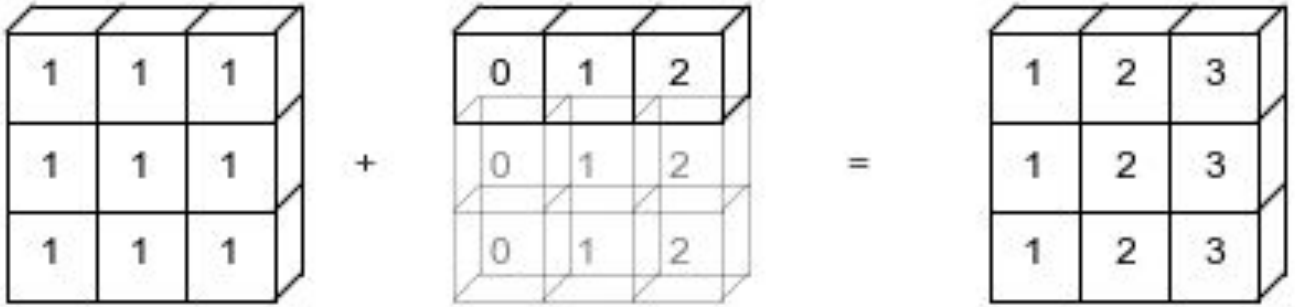
AI meets Industry

See Jupyter Notebook

Implementation	Elapsed Time
Pure Python with list comprehensions	18.65s
NumPy	0.32s
TensorFlow on CPU	1.20s

`np.arange(3)+5`

`np.ones((3,3))+np.arange(3)`



`np.arange(3).reshape((3,1))+np.arange(3)`