

# Python et Machine Learning

## TRAVAUX DIRIGÉS

### *Huitième partie*

Salim Lardjane

*Université de Bretagne Sud*

1. Ouvrir le programme *run\_ga.py* (disponible sur le Forum) sous Spyder, le lancer et le modifier de façon à ce qu'il fonctionne. Essayer de comprendre son fonctionnement, ainsi que celui de tous les programmes auxquels il fait appel.
2. Définir une classe *Domino()* qui permette d'instancier des objets simulant les pièces d'un jeu de dominos. Le constructeur de cette classe initialisera les valeurs des points présents sur les deux faces A et B du domino (valeurs par défaut = 0). Deux autres méthodes seront définies : une méthode *affiche\_points()* qui affiche les points présents sur les deux faces; une méthode *valeur()* qui renvoie la somme des points présents sur les deux faces.
3. Définir une classe *CompteBancaire()* qui permette d'instancier des objets tels que *compte1*, *compte2*, etc. Le constructeur de cette classe initialisera deux attributs d'instance *nom* et *solde*, avec les valeurs par défaut "Dupont" et 1000. Trois autres méthodes seront définies : *depot(somme)* permettra d'ajouter une certaine somme au solde; *retrait(somme)* permettra de retirer une certaine somme du solde; *affiche()* permettra d'afficher le nom du titulaire et le solde de son compte.
4. Définir une classe *Voiture()* qui permette d'instancier des objets reproduisant le comportement de voitures automobiles. Le constructeur de cette classe initialisera les attributs d'instance suivants, avec les valeurs par défaut indiquées : *marque* = 'Ford', *couleur* = 'rouge', *pilote* = 'personne', *vitesse* = 0. Lorsqu'on instanciera un nouvel objet *Voiture()*, on pourra choisir sa marque et sa couleur mais pas sa vitesse, ni le nom de son conducteur. Les méthodes suivantes seront définies : *choix\_conducteur(nom)* permettra de désigner (ou changer) le nom du conducteur; *accelerer(taux, duree)*

permettra de faire varier la vitesse de la voiture. La variation de vitesse obtenue sera égale au produit : taux x duree. Des taux négatifs seront acceptés, ce qui permettra de décélérer. La variation de vitesse ne sera pas autorisée si le conducteur est 'personne'; `affiche_tout()` permettra de faire apparaître les propriétés présentes de la voiture, c'est-à-dire sa marque, sa couleur, le nom de son conducteur, sa vitesse.

5. Définir une classe `Satellite()` qui permette d'instancier des objets simulant des satellites artificiels, lancés dans l'espace autour de la Terre. Le constructeur de cette classe initialisera les attributs d'instance suivants, avec les valeurs par défaut indiquées : `masse = 100`, `vitesse = 0`. Lorsqu'oninstanciera un nouvel objet `Satellite()`, on pourra choisir son nom, sa masse et sa vitesse. Les méthodes suivantes seront définies : `impulsion(force, duree)` permettra de faire varier la vitesse du satellite. La variation de vitesse  $\Delta v$  subie par un objet de masse  $m$  soumis à l'action d'une force  $F$  pendant un temps  $t$  vaut

$$\Delta v = \frac{F \times t}{m}$$

`affiche_vitesse()` affichera le nom du satellite et sa vitesse courante; `energie()` renverra au programme appelant la valeur de l'énergie cinétique du satellite. Celle-ci est donnée par la formule :

$$E_c = \frac{m \times v^2}{2}.$$

6. Définir une classe `Cercle()`. Les objets construits à partir de cette classe seront des cercles de tailles variées. En plus de la méthode constructeur (qui utilisera un paramètre `rayon`), définir une méthode `surface()`, qui devra renvoyer la surface du cercle.

Définir ensuite une classe `Cylindre()` dérivée de la précédente. Le constructeur de cette nouvelle classe comportera les deux paramètres `rayon` et `hauteur`. Y ajouter une méthode `volume()` qui devra renvoyer le volume du cylindre. Celui-ci est donné par la surface de la section x la hauteur.

Définir une classe `Cone()`, qui devra dériver de la classe `Cylindre()` et dont le constructeur comportera lui aussi les deux paramètres `rayon` et `hauteur`. Cette nouvelle classe possèdera sa propre méthode `volume()`, laquelle devra renvoyer le volume du cône. Celui-ci est donné par le volume du cylindre correspondant divisé par 3.

7. Définir une classe `JeuDeCartes()` permettant d'instancier des objets dont le comportement soit similaire à celui d'un vrai jeu de cartes. La classe

devra comporter au moins les quatre méthodes suivantes :

– méthode constructeur : création et remplissage d’une liste de 52 éléments, qui sont eux-mêmes des tuples de 2 entiers. Cette liste de tuples contiendra les caractéristiques de chacune des 52 cartes. Pour chacune d’elles, il faut en effet mémoriser séparément un entier indiquant sa valeur (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, les quatre dernières valeurs étant celles des valet, dame, roi et as), et un autre entier indiquant la couleur de la carte (c’est-à-dire 3, 2, 1, 0 pour Coeur, carreau, Trèfle et Pique). Dans une telle liste, l’élément (11,2) désigne donc le valet de Trèfle, et la liste terminée doit être du type : [(2,0), (3,0), (4,0), ..., (12,3), (13,3), (14,3)].

– méthode nom\_carte( ) : cette méthode doit renvoyer, sous la forme d’une chaîne, l’identité d’une carte quelconque dont on lui a fourni le tuple descripteur en argument. Par exemple, l’instruction `print(jeu.nom_carte((14,3)))` doit provoquer l’affichage de : **As de pique**.

– méthode battre( ) : cette méthode sert à mélanger les éléments de la liste contenant les cartes, quel qu’en soit le nombre.

– méthode tirer( ) : lorsque cette méthode est invoquée, une carte est retirée du jeu. Le tuple contenant sa valeur et sa couleur est renvoyé au programme appelant. On retire toujours la première carte de la liste. Si cette méthode est invoquée alors qu’il ne reste plus aucune carte dans la liste, il faut alors renvoyer l’objet spécial **None** au programme appelant.

8. Complément de l’exercice précédent : définir deux joueurs A et B. Instancier deux jeux de cartes (un pour chaque joueur) et les mélanger. Ensuite, à l’aide d’une boucle, tirer 52 fois une carte de chacun des joueurs et comparer leurs valeurs. Si c’est la première des deux qui a la valeur la plus élevée, on ajoute un point au joueur A. Si la situation contraire se présente, on ajoute un point au joueur B. Si les deux valeurs sont égales, on passe au tirage suivant. Au terme de la boucle, comparer les comptes de A et B pour déterminer le gagnant.