Prédiction d'habitat du renard

Sujet n°11 — Corrigé

Question 1 *Distance euclidienne entre deux habitats*

```
def distance(habitat_1, habitat_2):
    s = 0
    s = s + (habitat_1['vegetation'] - habitat_2['vegetation'])**2
    s = s + (habitat_1['proximite_eau'] - habitat_2['proximite_eau'])**2
    s = s + (habitat_1['densite_urbaine'] - habitat_2['densite_urbaine'])**2
    s = s + (habitat_1['disponibilite_proies'] - habitat_2['disponibilite_proies'])**2
    return sqrt(s)
```

Question 2 *Distances à une liste d'habitats*

```
def distance_d_un_habitat(habitat, habitats):
    resultat = []
    for h in habitats:
        d = distance(habitat, h)
        resultat.append((d, h))
    return resultat
```

Question 3 *Test de distance_d_un_habitat*

```
res = distance_d_un_habitat(nouveau, zones_connues)
for t in res[:3]:
    print(t)
```

Résultat (3 premiers tuples) :

```
(7.211102550927978, {'vegetation': 9, 'proximite_eau': 6,
    'densite_urbaine': 0, 'disponibilite_proies': 4, 'presence_renard': True})
(8.660254037844387, {'vegetation': 10, 'proximite_eau': 5,
    'densite_urbaine': 9, 'disponibilite_proies': 10, 'presence_renard': False})
(5.196152422706632, {'vegetation': 8, 'proximite_eau': 5,
    'densite_urbaine': 1, 'disponibilite_proies': 6, 'presence_renard': False})
```

Question 4 *Correction de presence_renard*

Cause : `if distance['presence_renard']` indice un `float` (indice 0 du tuple) au lieu du dictionnaire `caracteristiques` (indice 1) → `TypeError: 'float' object is not subscriptable`.

Correction : remplacer `distance['presence_renard']` par `caracteristiques['presence_renard']`.

```
def presence_renard(k, habitat, habitats):
    habitats = k_plus_proches(k, habitat, habitats)
    n_renards = 0
    for habitat in habitats:
        distance = habitat[0]
        caracteristiques = habitat[1]
        if caracteristiques['presence_renard']:
            n_renards += 1
    return n_renards > k/2
```

Question 5 *Prédiction pour l'habitat nouveau*

```
>>> presence_renard(1, nouveau, zones_connues)
False   # 0 renard sur 1 voisin
>>> presence_renard(3, nouveau, zones_connues)
True    # 2 renards sur 3 voisins
>>> presence_renard(5, nouveau, zones_connues)
True    # 3 renards sur 5 voisins
>>> presence_renard(7, nouveau, zones_connues)
True    # 5 renards sur 7 voisins
>>> presence_renard(9, nouveau, zones_connues)
True    # 6 renards sur 9 voisins
```

Dès $k \geq 3$, la majorité des voisins héberge un renard : `nouveau` est **susceptible d'accueillir une population de renards**.