

TP qualité du vin

Algorithme kNN

Christophe Viroulaud





















Première - NSI

Algo 07



FIGURE 1 – Les propriétés chimiques d'un vin influencent grandement sur sa qualité.

LE GUIDE DES MILLÉSIMES

Millésime		2017	2016	2015	2014	2013	2012	2011	2010	2009	2008
 Valais		5	5	5	4	4	4	4	5	5	5
 Valais		4	5	4	4	4	4	4	5	5	4
 Vaud, Neuchâtel		5	5	5	4	4	4	5	5	5	5
 Vaud, Neuchâtel		4	4	4	4	4	4	5	4	5	4
 Genève		5	4	5	4	4	4	4	4	5	5
 Genève		4	4	5	4	4	4	4	4	5	4
 Suisse alémanique		4	4	4	3	4	4	5	4	5	4
 Tessin		5	4	5	3	5	5	5	4	5	3
 Bordeaux / FR		4	5	5	4	3	4	3	5	5	4
 Bourgogne / FR		4	5	5	4	4	3	3	4	5	4
 Côtes-du-Rhône / FR		4	4	5	4	5	4	3	4	4	3
 Toscane / IT		4	5	5	4	5	4	4	4	4	3
 Piémont / IT		4	5	5	5	4	4	4	5	4	4
 Espagne		4	4	5	4	5	4	4	5	4	4
 Portugal		4	4	5	3	4	4	5	4	5	5
 Californie / USA		4	5	5	5	5	5	3	4	4	4
 Argentine		5	4	4	3	4	4	4	5	4	4
 Chili		5	4	4	5	4	3	4	5	4	4
 Australie		5	5	5	4	5	5	3	5	3	4
 Afrique du Sud		5	4	5	4	4	4	4	3	5	4

5	Exceptionnel		À encaver
4	Très bon		À son apogée
3	Bon		À consommer prochainement
2	Moyen		
1	Médiocre		

Études des données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

FIGURE 2 – Les œnologues établissent des classements des vins.

Établir un algorithme de classement des vins en fonction de leur propriétés chimiques.

1. Études des données

2. Algorithme kNN

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

On dispose d'un jeu de données sur plusieurs propriétés de différents vins :

- ▶ fixed acidity
- ▶ volatile acidity
- ▶ citric acid
- ▶ residual sugar
- ▶ chlorides
- ▶ free sulfur dioxide
- ▶ total sulfur dioxide
- ▶ density
- ▶ pH
- ▶ sulphates
- ▶ alcohol

Études des
données

Algorithme kNN

Principe
Importation des données
Distance
Trier
Sélectionner
Programme principal

Observation

De plus chaque vin a obtenu une note (**quality**) entre 1 et 8. Les données sont donc **étiquetées**.

Activité 1 :

1. Télécharger et extraire l'annexe `winequality-red.zip` sur le site <https://cviroulaud.github.io>
2. Ouvrir le fichier `winequality-red.csv` avec un tableur pour observer les données.

Études des
données

Algorithme kNN

Principe
Importation des données
Distance
Trier
Sélectionner
Programme principal

Observation

Lors de l'étude des iris, 2 caractéristiques seulement (longueur et largeur des pétales) étaient observées. Il était donc possible de les représenter graphiquement. Les vins possèdent 11 propriétés différentes ; une représentation graphique est donc impossible.

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

On dispose des propriétés d'un vin :

```
1 vin_inconnu = {'fixed acidity': 6.9, 'volatile  
acidity': 0.5, 'citric acid': 0.19, 'residual  
sugar': 3.9, 'chlorides': 0.16, 'free sulfur  
dioxide': 31.0, 'total sulfur dioxide': 50.0, '  
density': 0.994, 'pH': 3.01, 'sulphates': 0.61, '  
alcohol': 9.3}
```

Et on cherche à déterminer une note (**quality**) en établissant un modèle dans le jeu de données fourni.

L'algorithme est similaire à celui utilisé pour les iris :

- ▶ Charger les données dans le programme.
- ▶ Choisir k .
- ▶ Stocker les propriétés du vin inconnu.
- ▶ Calculer la distance euclidienne entre le vin inconnu et tous les autres vins.
- ▶ Trier les vins selon leurs notes.
- ▶ Sélectionner les k plus proches vin (en distance) du vin inconnu.
- ▶ Affecter la note majoritaire des k plus proches vins (en distance) au vin inconnu.

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Activité 2 :

1. Créer le fichier `qualitevin.py` dans le même dossier que le fichier `csv`.
2. Importer les données des vins dans le programme.
3. Créer un tableau `vins` de dictionnaires. Chaque dictionnaire représentera un vin du fichier `csv`. Toutes les propriétés seront converties en nombre flottant sauf `quality` qui sera converti en entier.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 fichier = open(f, encoding="utf8")
2 data = csv.DictReader(fichier)
3
4 vins = []
5 for v in data:
6     vin = {}
7     for attribut, valeur in v.items():
8         # qualité est le seul entier
9         if attribut == "quality":
10             vin[attribut] = int(valeur)
11         else:
12             vin[attribut] = float(valeur)
13     vins.append(vin)
14
15 fichier.close()
```

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Même si les données ne sont pas représentables graphiquement, il est possible de calculer la distance euclidienne entre deux vins :

$$\begin{aligned} \textit{distance} = & \\ & (\textit{fixedacidity}_{\textit{connu}} - \textit{fixedacidity}_{\textit{inconnu}})^2 + \\ & (\textit{volatileacidity}_{\textit{connu}} - \textit{volatileacidity}_{\textit{inconnu}})^2 + \\ & (\textit{citricacid}_{\textit{connu}} - \textit{citricacid}_{\textit{inconnu}})^2 + \\ & \dots \end{aligned}$$

Activité 3 : Écrire la fonction `distance(connu: dict, inconnu: dict) → float` qui calcule le carré de la distance euclidienne entre un vin dont la note (`quality`) est connue et un dont la note est inconnue.
Remarque : Le dictionnaire du vin inconnu ne possède donc pas de clé `quality`.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 def distance(connu: dict, inconnu: dict) -> float:
2     """
3     calcule (le carré de) la distance euclidienne
4     entre les vins connu et inconnu
5     """
6     dist = 0
7     for attribut, valeur in connu.items():
8         # inconnu n'a pas d'attribut "quality" (c'est ce
9         qu'on veut déterminer)
10        if not(attribut == "quality"):
11            dist += (connu[attribut]-inconnu[attribut])**2
12    return dist
```

tion des données

ner

me principal

Activité 4 : Écrire la fonction `calculer_distances(vins: list, inconnu: dict) → list` qui renvoie un tableau de tuples. Chaque tuple contiendra la note d'un vin connu et sa distance au vin inconnu.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 def calculer_distances(vins: list, inconnu: dict) -> list:
2     distances = []
3     for v in vins:
4         d = distance(v, inconnu)
5         # stocke le tuple (qualité, distance)
6         distances.append((v["quality"], d))
7
8     return distances
```

onner
me principal

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Activité 5 :

1. Créer le fichier `tri.py` dans le même dossier.
2. Copier le tri par insertion vu en classe dans ce fichier.
3. Les éléments à trier sont des tuples :

```
1 # quality, distance  
2 (5, 5.166)
```

Modifier la fonction `insérer` pour trier les tuples en fonction de la distance (second élément du tuple).

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 def echanger(tab: list, i: int, j: int) -> None:
2     temp = tab[i]
3     tab[i] = tab[j]
4     tab[j] = temp
5
6 def inserer(tab: list, j: int) -> None:
7     """
8     tri en fonction du second élément du tuple
9     """
10    # Le changement se fait dans la comparaison
11    while j-1 >= 0 and tab[j-1][1] > tab[j][1]:
12        echanger(tab, j-1, j)
13        j = j-1
14
15 def tri_insertion(tab: list) -> None:
16     for i in range(len(tab)):
17         inserer(tab, i)
```

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Activité 6 : Écrire la fonction `trouver_qualite(k: int, distances: list) → int` qui renvoie la note (`quality`) la plus présente parmi les `k` premiers tuples du tableau `distances`.

La fonction :

- ▶ construira un dictionnaire `compteur_qualites` qui associera chaque note `quality` à son nombre d'apparition parmi les `k` premiers tuples,
- ▶ sélectionnera la note du dictionnaire qui est associée à la plus grande valeur,
- ▶ renverra cette note.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 def trouver_qualite(k: int, distances: list) -> int:
2     # construire le dictionnaire
3     compteur_qualites = {}
4     for i in range(k):
5         qualite = distances[i][0]
6         if qualite in compteur_qualites:
7             compteur_qualites[qualite] += 1
8         else:
9             compteur_qualites[qualite] = 1
10    # sélectionner la note la plus présente
11    maxi = 0
12    qualite_maxi = 0
13    for qualite, valeur in compteur_qualites.items():
14        if valeur > maxi:
15            qualite_maxi = qualite
16            maxi = valeur
17    # renvoyer la note
18    return qualite_maxi
```

1. Études des données

2. Algorithme kNN

2.1 Principe

2.2 Importation des données

2.3 Distance

2.4 Trier

2.5 Sélectionner

2.6 Programme principal

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Programme principal

Il reste à utiliser les fonctions précédentes pour appliquer l'algorithme :

- ▶ Charger les données dans le programme.
- ▶ Choisir k .
- ▶ Stocker les propriétés du vin inconnu.
- ▶ Calculer la distance euclidienne entre le vin inconnu et tous les autres vins.
- ▶ Trier les vins selon leurs notes.
- ▶ Sélectionner les k plus proches vin (en distance) du vin inconnu.
- ▶ Affecter la note majoritaire des k plus proches vins (en distance) au vin inconnu.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

Activité 7 : Écrire le programme principal qui implémente l'algorithme précédent. On utilisera :

```
1 k = 3
2 vin_inconnu = {'fixed acidity': 6.9, 'volatile
  acidity': 0.5, 'citric acid': 0.19, 'residual
  sugar': 3.9, 'chlorides': 0.16, 'free sulfur
  dioxide': 31.0, 'total sulfur dioxide':
  50.0, 'density': 0.994, 'pH': 3.01, '
  sulphates': 0.61, 'alcohol': 9.3}
```

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Études des
données

Algorithme kNN

Principe

Importation des données

Distance

Trier

Sélectionner

Programme principal

```
1 k = 3
2 vin_inconnu = {'fixed acidity': 6.9, 'volatile
  acidity': 0.5, 'citric acid': 0.19, 'residual
  sugar': 3.9, 'chlorides': 0.16, 'free sulfur
  dioxide': 31.0, 'total sulfur dioxide': 50.0, '
  density': 0.994, 'pH': 3.01, 'sulphates': 0.61, '
  alcohol': 9.3}
3
4 distances = calculer_distances(vins, vin_inconnu)
5 tri_insertion(distances)
6 print(trouver_qualite(k, distances))
```