

mettre iris.zip sur site

Prédire la variété d'un iris

Christophe Viroulaud

Première - NSI

Prédire la variété d'un iris

Christophe Viroulaud

Première - NSI

1. Problématique
2. Utiliser les données
3. Algorithme kNN

Sommaire

1. Problématique

2. Utiliser les données

3. Algorithme kNN

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Prédire la variété d'un iris

Problématique

Il est possible d'utiliser ces données pour pouvoir classifier un iris inconnu.
 Comment produire déduction à partir de données existantes ? → prémisse
 IA : machine learning

En 1936, le biologiste *Ronald Fisher* a rassemblé les mesures de trois espèces d'iris.



Iris setosa Iris versicolor Iris virginica

En 1936, le biologiste *Ronald Fisher* a rassemblé les mesures de trois espèces d'iris.



Iris setosa



Iris versicolor



Iris virginica

Comment prédire une information nouvelle à partir de données brutes ?

Comment prédire une information nouvelle à partir de données brutes ?

Problématique

Utiliser les données

Présentation graphique des informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Sommaire

1. Problématique

2. Utiliser les données

2.1 Présentation graphique des informations

2.2 Prédire la variété

3. Algorithme kNN

Prédire la variété d'un iris

Utiliser les données

Présentation graphique des informations

Présentation graphique des informations

1. Une représentation graphique des informations apporte une compréhension plus éclairante.
2. Il apparaît que les mesures d'un iris peuvent permettre de déterminer leur variété.

Présentation graphique des informations

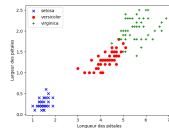


FIGURE 1 – Variétés d'iris en fonction de leurs mesures

Présentation graphique des informations

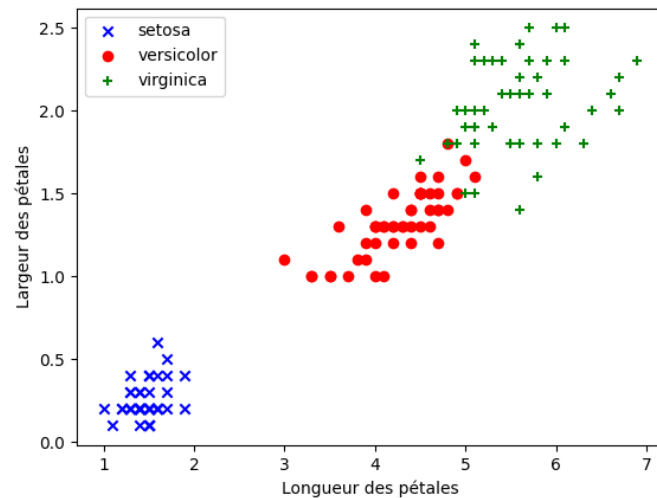


FIGURE 1 – Variétés d'iris en fonction de leurs mesures

Sommaire

1. Problématique

2. Utiliser les données

2.1 Présentation graphique des informations

2.2 Prédire la variété

3. Algorithme kNN

Prédire la variété d'un iris

- Utiliser les données
 - Prédire la variété
 - Prédire la variété

Activité 1 :

1. Déterminer la variété des iris suivants :

longueur	1	6	5.1	2.5
largeur	0.5	2.5	1.55	0.85

2. Proposer une méthode pour effectuer un choix dans les cas ambigus.

Prédire la variété

Activité 1 :

- Déterminer la variété des iris suivants :

longueur	1	6	5.1	2.5
largeur	0.5	2.5	1.55	0.85

- Proposer une méthode pour effectuer un choix dans les cas ambigus.

Prédire la variété d'un iris

└ Utiliser les données

└ Prédire la variété

└ Avant de regarder la correction

Avant de regarder la correction



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

Avant de regarder la correction



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

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Prédire la variété d'un iris

- Utiliser les données
 - Prédire la variété
 - Correction

Correction

longueur	1	6	5.1	2.5
largeur	0.5	2.5	1.55	0.85
variété	setosa	virginica	ambigu	ambigu

Correction

longueur	1	6	5.1	2.5
largeur	0.5	2.5	1.55	0.85
variété	setosa	virginica	ambigu	ambigu

Prédire la variété d'un iris

Problématique

Utiliser les données

Présentation graphique des informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Sommaire

1. Problématique

2. Utiliser les données

3. Algorithme kNN

3.1 Présentation

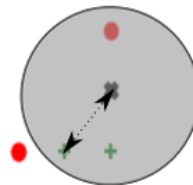
3.2 Construction de l'algorithme

3.3 Implémentation

k Nearest NeighborsMéthode des **k plus proches voisins**

Pour déterminer la variété d'un iris inconnu :

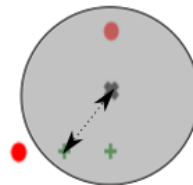
- regarder la variété d'un nombre k de voisins,



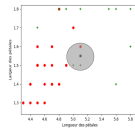
k Nearest NeighborsMéthode des **k plus proches voisins**

Pour déterminer la variété d'un iris inconnu :

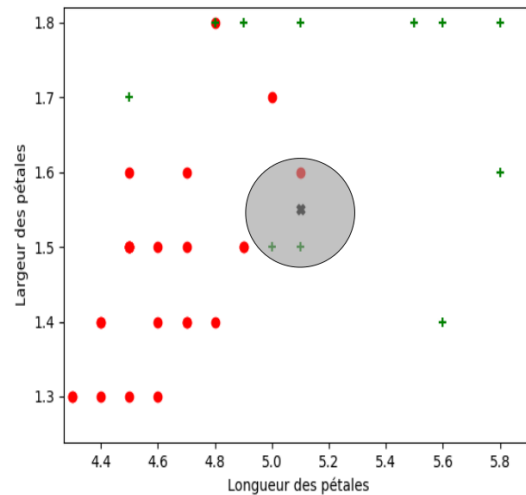
- regarder la variété d'un nombre k de voisins,



- attribuer à la fleur inconnue, la variété la plus présente
parmi ses k voisins.

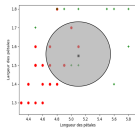
FIGURE 2 – Détermination de l'iris (5.05, 1.5) pour $k = 3$

Choix de k

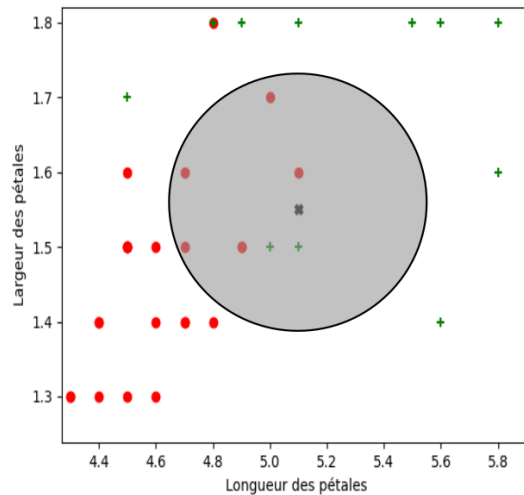
FIGURE 2 – Détermination de l'iris (5.05, 1.5) pour $k = 3$

1. avantages de kNN : simple d'implémentation et résultats avec bon taux de réussite quand on a un gros échantillon
2. inconvénients : sensible au bruit (données mal étiquetées), sensible à l'échelle de chaque dimension

Choix de k

FIGURE 3 – Détermination de l'iris (5.05, 1.5) pour $k = 7$

Choix de k

FIGURE 3 – Détermination de l'iris (5.05, 1.5) pour $k = 7$

Complément

L'algorithme *kNN* est une méthode d'apprentissage *supervisé* : l'algorithme reçoit un ensemble de données déjà étiquetées sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction.

1. non supervisé : données pas étiquetées
2. par renforcement : par récompense

Complément

L'algorithme *kNN* est une méthode d'apprentissage *supervisé* : l'algorithme reçoit un ensemble de données déjà étiquetées sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction.

Sommaire

1. Problématique

2. Utiliser les données

3. Algorithme kNN

3.1 Présentation

3.2 Construction de l'algorithme

3.3 Implémentation

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Construction de l'algorithme

└─ Calcul de la distance

Calcul de la distance

Le plus naturel ici est de prendre la distance à vol d'oiseau ou plus formellement la **distance euclidienne**.

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$



FIGURE 4 – distance euclidienne

Calcul de la distance

Le plus naturel ici est de prendre la distance à *vol d'oiseau* ou plus formellement la **distance euclidienne**.

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

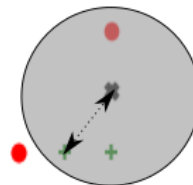


FIGURE 4 – distance euclidienne

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Construction de l'algorithme

└─ Calcul de la distance

Calcul de la distance

$$d = |x_A - x_B| + |y_A - y_B|$$



FIGURE 5 – distance de Manhattan

Calcul de la distance

$$d = |x_A - x_B| + |y_A - y_B|$$

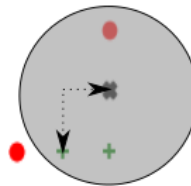


FIGURE 5 – distance de Manhattan

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Activité 2 : Écrire *en langage naturel*, l'algorithme kNN.

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Construction de l'algorithme

└─ Avant de regarder la correction

Avant de regarder la correction



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

Avant de regarder la correction



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

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Construction de l'algorithme

└─ Correction

Correction

- ▶ Charger les données dans le programme.
- ▶ Choisir k .
- ▶ Stocker les mesures de la fleur inconnue.
- ▶ Calculer la distance euclidienne entre la fleur inconnue et tous les autres iris.
- ▶ Sélectionner les k plus proches iris (en distance) de la fleur inconnue.
- ▶ Affecter la variété majoritaire des k plus proches iris (en distance) à la fleur inconnue.

Correction

- ▶ Charger les données dans le programme.
- ▶ Choisir k .
- ▶ Stocker les mesures de la fleur inconnue.
- ▶ Calculer la distance euclidienne entre la fleur inconnue et tous les autres iris.
- ▶ Sélectionner les k plus proches iris (en distance) de la fleur inconnue.
- ▶ Affecter la variété majoritaire des k plus proches iris (en distance) à la fleur inconnue.

Sommaire

1. Problématique

2. Utiliser les données

3. Algorithme kNN

3.1 Présentation

3.2 Construction de l'algorithme

3.3 Implémentation

Pour charger les données on utilisera la bibliothèque csv.

Pour charger les données on utilisera la bibliothèque csv.

Prédire la variété d'un iris

- └─ Algorithme kNN
 - └─ Implémentation

Activité 3 :

1. Télécharger le dossier compressé *iris.zip* sur le site <https://cviroulaud.github.io>
2. Ouvrir le fichier *data-iris.csv* avec un tableur pour observer les données.
3. Ouvrir le fichier *iris-eleve.py*.

Activité 3 :

1. Télécharger le dossier compressé *iris.zip* sur le site <https://cviroulaud.github.io>
2. Ouvrir le fichier *data-iris.csv* avec un tableur pour observer les données.
3. Ouvrir le fichier *iris-eleve.py*.



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

Avant de regarder la correction



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

petal_length	petal_width	species
1.4	0.2	setosa
1.4	0.2	setosa
1.3	0.2	setosa

Correction

3 attributs

petal_length	petal_width	species
1.4	0.2	setosa
1.4	0.2	setosa
1.3	0.2	setosa

Prédire la variété d'un iris

- Algorithme kNN
 - Implémentation

Activité 3 :

4. Compléter la fonction `charger_donnees` en utilisant les informations du fichier `csv`.
5. Compléter la fonction `distance` qui calcule le carré de la distance euclidienne entre deux points du plan.
6. Compléter la fonction `calculer_distances`.
7. Compléter enfin la fonction `trouver_variete`. Le dictionnaire `compteur_voisins` compte le nombre d'apparitions de chaque variété parmi les k voisins.

Activité 3 :

4. Compléter la fonction `charger_donnees` en utilisant les informations du fichier `csv`.
5. Compléter la fonction `distance` qui calcule le carré de la distance euclidienne entre deux points du plan.
6. Compléter la fonction `calculer_distances`.
7. Compléter enfin la fonction `trouver_variete`. Le dictionnaire `compteur_voisins` compte le nombre d'apparitions de chaque variété parmi les k voisins.

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation



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

Avant de regarder la correction



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

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Implémentation

└─ Correction

Correction

```

1 def charger_donnees(nom_fichier: str) -> dict:
2     fichier = open(nom_fichier)
3     data_iris = csv.DictReader(fichier, delimiter=",")
4     dico_varietes = {"setosa": [], "versicolor": [], "virginica":
5         []}
6     # Pour chaque ligne de données
7     for iris in data_iris:
8         # Stocke la longueur et la largeur sous forme de tuple
9         # de flottants
10        dico_varietes[iris["species"]].append(
11            (float(iris["petal_length"]), float(iris["petal_width"])))
12    fichier.close()
13    return dico_varietes

```

Correction

```

1 def charger_donnees(nom_fichier: str) -> dict:
2     fichier = open(nom_fichier)
3     data_iris = csv.DictReader(fichier, delimiter=",")
4     dico_varietes = {"setosa": [], "versicolor": [], "virginica":
5         []}
6     # Pour chaque ligne de données
7     for iris in data_iris:
8         # Stocke la longueur et la largeur sous forme de tuple
9         # de flottants
10        dico_varietes[iris["species"]].append(
11            (float(iris["petal_length"]), float(iris["petal_width"])))
12    fichier.close()
13    return dico_varietes

```

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

```
1 def distance(connu: tuple, inconnu: tuple) -> float:
2     return (connu[0]-inconnu[0])**2+(connu[1]-inconnu[1])
   **2
```

Correction

```
1 def distance(connu: tuple, inconnu: tuple) -> float:
2     return (connu[0]-inconnu[0])**2+(connu[1]-inconnu[1])
   **2
```

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Implémentation

└─ Correction

Correction

```

1 def calculer_distances(donnees: dict, inconnu: tuple) -> list:
2     distances = []
3     for nom, mesures in donnees.items():
4         for iris in mesures:
5             d = distance(iris, inconnu)
6             distances.append((nom, d))
7
8     # trie les iris en fonction de la distance
9     distances.sort(key=lambda fleur: fleur[1])
10    return distances

```

Correction

```

1 def calculer_distances(donnees: dict, inconnu: tuple) -> list:
2     distances = []
3     for nom, mesures in donnees.items():
4         for iris in mesures:
5             d = distance(iris, inconnu)
6             distances.append((nom, d))
7
8     # trie les iris en fonction de la distance
9     distances.sort(key=lambda fleur: fleur[1])
10    return distances

```

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Implémentation

└─ Correction

Correction

```

1 def trouver_variete(k: int, distances: list) -> str:
2     # compte le nombre d'occurrences de chaque variété
3     compteur_voisins = {}
4     for i in range(k):
5         nom = distances[i][0]
6         if nom in compteur_voisins:
7             compteur_voisins[nom] += 1
8         else:
9             compteur_voisins[nom] = 1
10
11     # recherche la variété avec la plus grande valeur dans
12     # compteur_voisins
13     maxi = 0
14     nom_maxi = 0
15     for nom, quantite in compteur_voisins.items():
16         if quantite > maxi:
17             maxi = quantite
18             nom_maxi = nom
19     return nom_maxi

```

Correction

```

1 def trouver_variete(k: int, distances: list) -> str:
2     # compte le nombre d'occurrences de chaque variété
3     compteur_voisins = {}
4     for i in range(k):
5         nom = distances[i][0]
6         if nom in compteur_voisins:
7             compteur_voisins[nom] += 1
8         else:
9             compteur_voisins[nom] = 1
10
11     # recherche la variété avec la plus grande valeur dans
12     # compteur_voisins
13     maxi = 0
14     nom_maxi = 0
15     for nom, quantite in compteur_voisins.items():
16         if quantite > maxi:
17             maxi = quantite
18             nom_maxi = nom
19     return nom_maxi

```

Prédire la variété d'un iris

- └─ Algorithme kNN
 - └─ Implémentation

Activité 3 :

- 8. Tester la fonction avec $k = 3$ puis $k = 7$, puis pour les autres iris de l'activité 1.
- 9. Pour les plus avancés : Modifier le code pour tester un ensemble de 10 iris inconnus. De plus chaque iris déterminé sera ajouté au dictionnaire *varietes* afin d'augmenter l'apprentissage de l'algorithme.

Activité 3 :

- 8. Tester la fonction avec $k = 3$ puis $k = 7$, puis pour les autres iris de l'activité 1.
- 9. Pour les plus avancés : Modifier le code pour tester un ensemble de 10 iris inconnus. De plus chaque iris déterminé sera ajouté au dictionnaire *varietes* afin d'augmenter l'apprentissage de l'algorithme.



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

Avant de regarder la correction



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

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Implémentation

└─ Correction

Correction

```
1 k = 3
2 cible = (5.1, 1.55)
3
4 varietes = charger_donnees("data-iris.csv")
5 distances_cible = calculer_distances(varietes, cible)
6 variete = trouver_variete(k, distances_cible)
7
8 print("La variété est ", variete)
```

Correction

```
1 k = 3
2 cible = (5.1, 1.55)
3
4 varietes = charger_donnees("data-iris.csv")
5 distances_cible = calculer_distances(varietes, cible)
6 variete = trouver_variete(k, distances_cible)
7
8 print("La variété est ", variete)
```

Prédire la variété
d'un iris

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Prédire la variété d'un iris

└─ Algorithme kNN

└─ Implémentation

└─ Correction

Correction

```

1 k = 3
2 cibles = [(1,0.5),(6,2.5),(5.1, 1.55),(2.5,0.85),(3,2),
3           (6,1.2),(2,1,1),(3,2,1.5),(3,5,2.5),(4,1)]
4
5 varietes = charger_donnees("data-iris.csv")
6 for iris_inconnu in cibles:
7     # trouve la variété
8     distances_cible = calculer_distances(varietes,
9     iris_inconnu)
10    variete = trouver_variete(k, distances_cible)
11    print(f"La variété de {iris_inconnu} est {variete}.")
12    # ajout de cible au dictionnaire des données
13    varietes[variete].append(iris_inconnu)

```

Correction

```

1 k = 3
2 cibles = [(1,0.5),(6,2.5),(5.1, 1.55),(2.5,0.85),(3,2),
3           (6,1.2),(2,1,1),(3,2,1.5),(3,5,2.5),(4,1)]
4
5 varietes = charger_donnees("data-iris.csv")
6 for iris_inconnu in cibles:
7     # trouve la variété
8     distances_cible = calculer_distances(varietes,
9     iris_inconnu)
9     variete = trouver_variete(k, distances_cible)
10    print(f"La variété de {iris_inconnu} est {variete}.")
11    # ajout de cible au dictionnaire des données
12    varietes[variete].append(iris_inconnu)

```

Problématique

Utiliser les données

Présentation graphique des
informations

Prédire la variété

Algorithme kNN

Présentation

Construction de l'algorithme

Implémentation

Code complet

Le code complet est accessible [ici](#).