ESMT
École Supérieure Multinationale des Télécommunications
BÉNIN BURKINA FASO GUINÉE MALI MAURITANIE NIGER SÉNÉGAL

MACHINE LEARING

K-Nearest Neighbour

Habib Aidara

Mamadou Ba

Moussa .S. Sanogo

Mr Tondji

SOMAIRE

I. INTRODUCTION

En apprentissage supervisé, un algorithme reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction. Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes, d'où le K-NN l'un parmi ces algorithmes.

1. Définition du K-NN

L'algorithme des K plus proches voisins ou K-Nearest Neighbors (K-NN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression.

II. PRINCIPE ET PREDICTION

1. Principe

Son fonctionnement peut être assimilé à l'analogie suivante : ***"dis-moi qui sont tes voisins, je te dirais qui tu es…".***
L'algorithme KNN à la phase d'entraînement stocke simplement le jeu de données et lorsqu'il obtient de nouvelles données, il classe ces données dans une catégorie très similaire aux nouvelles données.

Exemple : Supposons que nous ayons une image d'une créature qui ressemble à un chat et à un chien, mais nous voulons savoir s'il s'agit d'un chat ou d'un chien. Donc, pour cette identification, nous pouvons utiliser l'algorithme KNN, car il fonctionne sur une mesure de similitude. Notre modèle KNN trouvera les caractéristiques similaires du nouvel ensemble de données aux images de chats et de chiens et, sur la base des caractéristiques les plus similaires, il le placera dans la catégorie des chats ou des chiens.

2. Prédiction

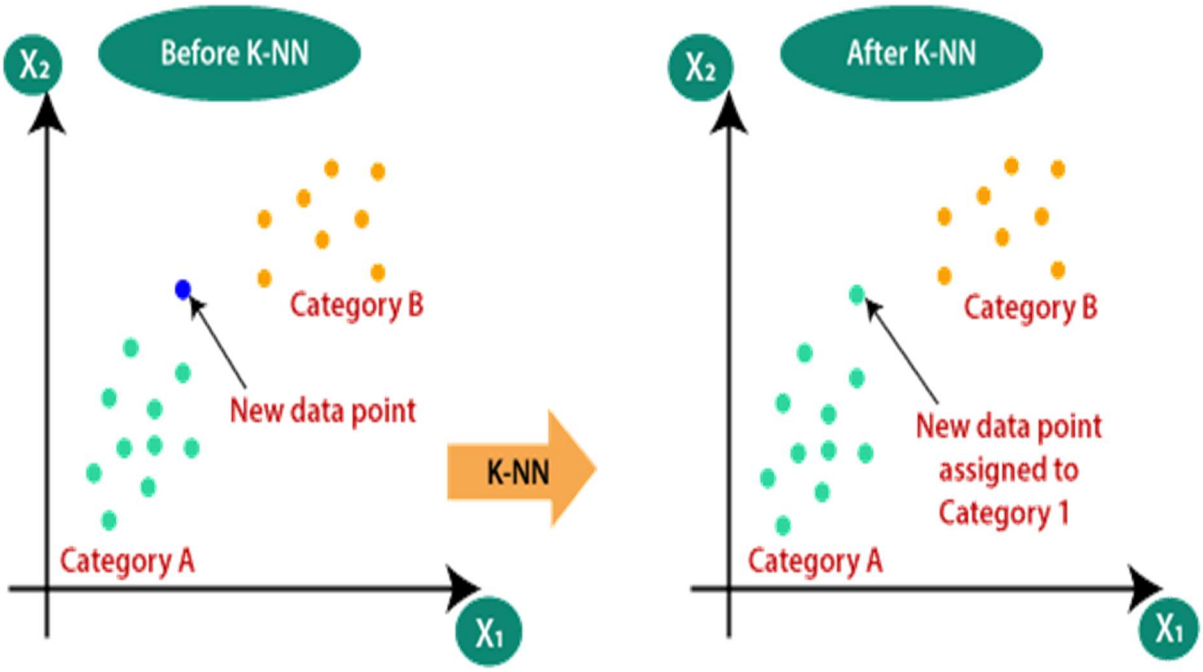
Comment le K-NN effectue une prédiction ?

Pour effectuer une prédiction, l'algorithme K-NN va se baser sur le jeu de données en entier. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les K instances du jeu de données les plus proches de notre observation. Ensuite pour ces voisins, l'algorithme se basera sur leurs variables de sortie (output variable) pour calculer la valeur de la variable de l'observation qu'on souhaite prédire.

Par ailleurs :

- Si K-NN est utilisé pour la régression, c'est la moyenne (ou médiane) des variables des plus proches observations qui servira pour la prédiction
- Si K-NN est utilisé pour la classification, c'est le mode des variables des plus proches observations qui servira pour la prédiction.

EXEMPLE : Supposons qu'il y ait deux catégories, c'est-à-dire la catégorie A et la catégorie B, et que nous ayons un nouveau point de données x1, de sorte que ce point de données se trouvera dans laquelle de ces catégories. Pour résoudre ce type de problème, nous avons besoin d'un algorithme K-NN. Avec l'aide de K-NN, nous pouvons facilement identifier la catégorie ou la classe d'un ensemble de données particulier. Considérez le diagramme ci-dessous :



III. ALGORITHME

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant :

Début Algorithme

Données en entrée :

- Un ensemble de données D
- Une fonction de définition distance d.
- Un nombre entier K

Pour une nouvelle observation dont on veut prédire sa variable de sortie Faire :

1. Calculer toutes les distances de cette observation avec les autres observations du jeu de données
2. Retenir les observations du jeu de données les proches en utilisant la fonction de calcul de distance.
3. Prendre les valeurs de des observations retenues :
    a. Si on effectue une régression, calculer la moyenne (ou la médiane) de retenues
    b. Si on effectue une classification, calculer le mode de retenues
4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation.

Fin Algorithme.

1. Calcul de similarité dans l'algorithme du K-NN

L'algorithme, K-NN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires et vice versa. Il existe plusieurs fonctions de calcul de distance :

- **La distance euclidienne**
- **La distance de Manhattan**
- **La distance de Minkowski**
- **La distance Jaccard,**
- **La distance de Hamming**…etc.

On choisit la fonction de distance en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple : poids, salaires, taille, montant de panier électronique etc…) et du même type, la **distance euclidienne** est un bon candidat. Quant à la distance de **Manhattan**, elle est une bonne mesure à utiliser quand les données (input variables) ne sont pas du même type (exemple : âge, sexe, longueur, poids etc…).

a) Distance Euclidienne :

Distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(X, Y) = \sqrt{\sum_{j=1}^{N} (X_j - Y_j)^2}$$

b) Distance Manhattan

La distance de Manhattan est la somme des valeurs absolues des différences entre les coordonnées de deux points

$$D_m(X,Y) = \sum_{i=1}^{k} |X_j - Y_j|$$

c) Distance de Hamming

La distance entre deux points donnés est la différence maximale entre leurs coordonnées sur une dimension :

$$D_m(X,Y) = \sum_{i=1}^{k} |X_j - Y_j|$$

- X = Y $\rightarrow$ D = 1
- X $\neq$ Y $\rightarrow$ D = 0

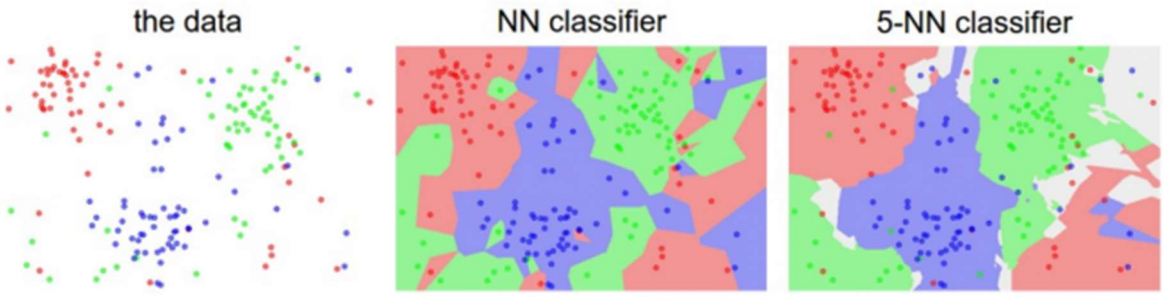
Il existe d'autres distances selon le cas d'utilisation de l'algorithme, mais la **distance euclidienne** reste la plus utilisée.

2. Choix de la valeur de K

Le choix de la valeur à utiliser pour effectuer une prédiction avec K-NN, varie en fonction du jeu de données. En règle générale, moins on utilisera de voisins (un nombre petit) plus on sera sujette au sous apprentissage (underfitting).

Par ailleurs, plus on utilise de voisins (un nombre K grand) plus, sera fiable dans notre prédiction.
Toutefois, si on utilise nombre de voisins avec et étant le nombre d'observations, on risque d'avoir du overfitting et par

conséquent un modèle qui se généralise mal sur des observations qu'il n'a pas encore vu.

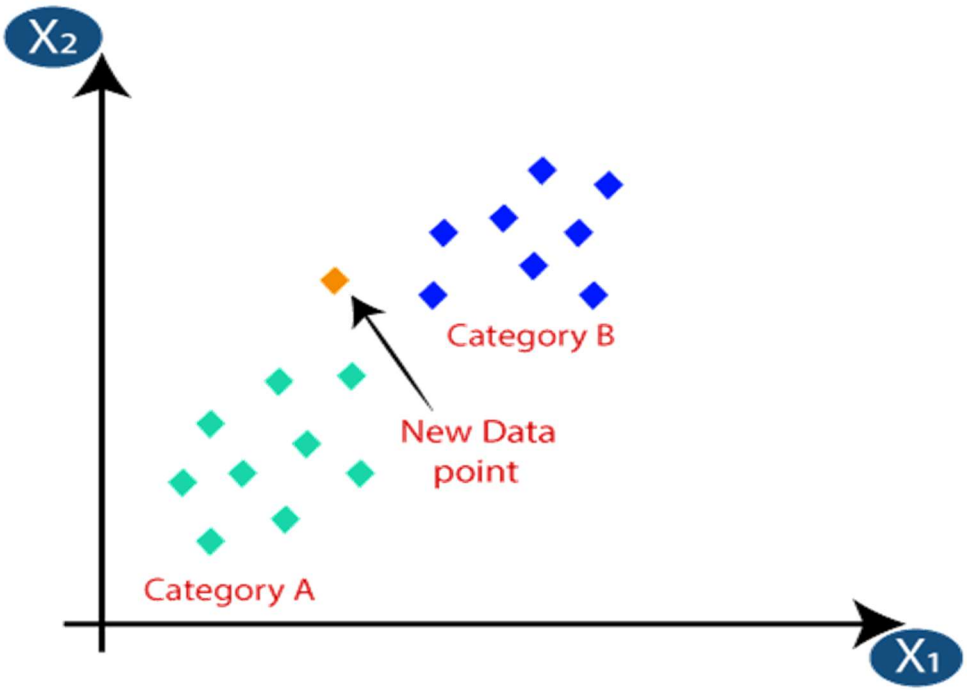


L'image ci-dessus à gauche représente des points dans un plan 2D avec trois types d'étiquetages possibles (rouge, vert, bleu). Pour le 5-NN classifieur, les limites entre chaque région sont assez lisses et régulières. Quant au N-NN Classifier, on remarque que les limites sont "chaotiques" et irrégulières. Cette dernière provient du fait que l'algorithme tente de faire rentrer tous les points bleus dans les régions bleues, les rouges avec les rouges etc… c'est un cas d'overfitting.

Pour cet exemple, on préférera le 5-NN classifier sur le NN-Classifier. Car le 5-NN classifier se généralise mieux que son opposant.

Exemple :

Supposons que nous ayons un nouveau point de données et que nous devions le mettre dans la catégorie requise. Considérez l'image ci-dessous :



Tout d'abord, nous choisirons le nombre de voisins, nous choisirons donc le k = 5.

Ensuite, nous allons calculer la **distance euclidienne** entre les points de données.

Comme nous pouvons le voir, les 3 voisins les plus proches sont de la catégorie A, donc ce nouveau point de données doit appartenir à la catégorie A.

IV. LES LIMITES DU K-NN

Avantages de l'algorithme KNN

- C'est simple à mettre en œuvre.
- Il est robuste aux données d'entraînement bruyantes
- Cela peut être plus efficace si les données d'entraînement sont volumineuses.

Inconvénients de l'algorithme K-NN:

- Il faut toujours déterminer la valeur de K qui peut être complexe à un moment donné.
- Le coût de calcul est élevé en raison du calcul de la distance entre les points de données pour tous les échantillons d'entraînement.

Implémentation du KNN AVEC SCIKIT_LEARN

++++++++++++++++++++++++++++++++++++

V. CONCLUSION

- K-NN stocke tout le jeu de données pour effectuer une prédiction.
- K-NN ne calcule aucun modèle prédictif et il rentre dans
- le cadre du *Lazy Learning*,
- K-NN effectue des prédictions justes à temps (à la volée) en calculant la similarité entre un observation en entrée et les différentes observations du jeu de données.

MERCI DE VOTRE ATTENTION !!!