Aix Marseille Université - Campus Luminy

UFR des Sciences

Rapport de TP

Master Informatique

Module ISD : Introduction à la Science des Données.

TP n°2 :

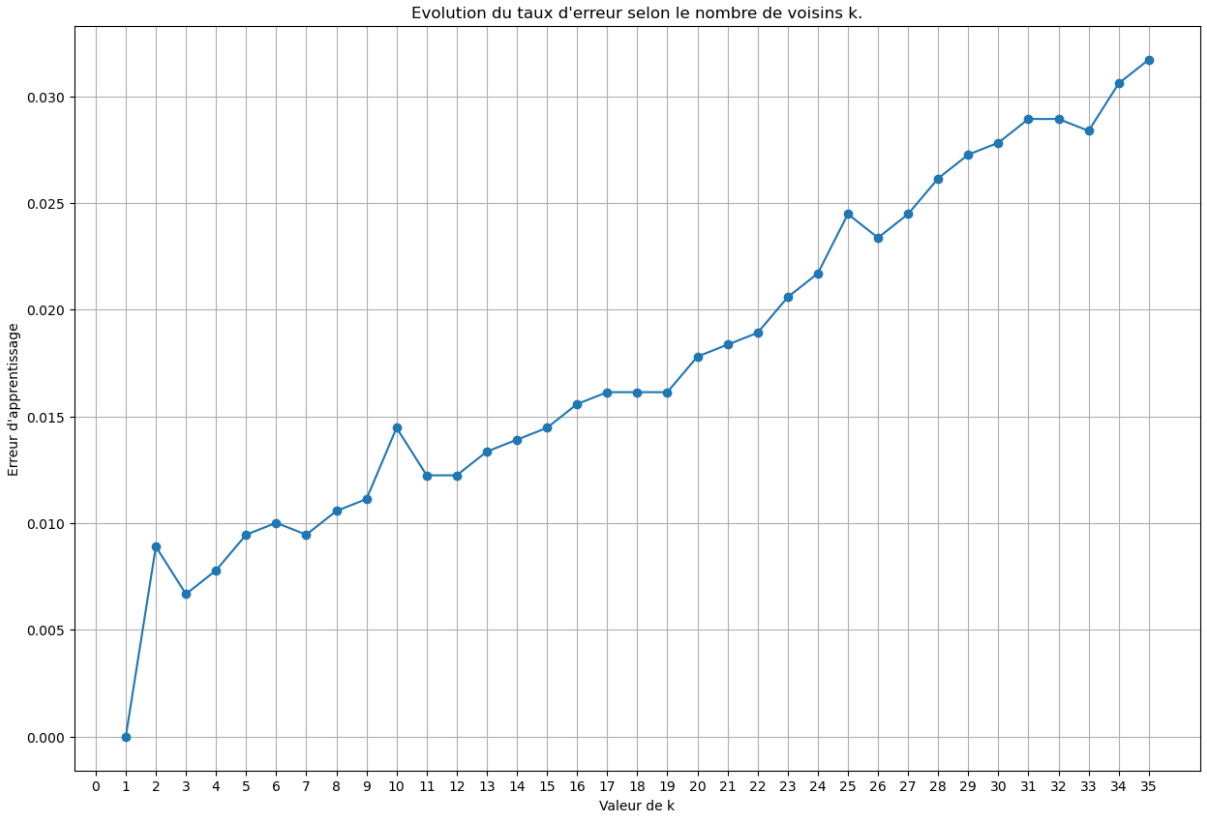
Classification par les k plus proches voisins.

Réalisé par :

ZEMMOURI Yasmine G3.

Partie 3 : Variation du nombre de voisins :

- Evolution de l'erreur d'apprentissage en fonction de la variation de l'hyper-paramètre k, nombre de voisins, du classifieur KNN (Kppv) :



✔ Lors de l'observation de la courbe, on remarque que l'erreur d'apprentissage augmente à mesure que k augmente.

✔ On observe que pour k=1, l'erreur d'apprentissage est nulle, car chaque exemple est associé à lui-même (chaque 1-voisin est lui-même), sa propre classe lui est attribuée. On constate ainsi que k=1 a tendance à surapprendre les données d'apprentissage, conduisant à une erreur d'apprentissage nulle. Cela ne devient plus vrai si l'on estime l'erreur de l'algorithme avec un autre échantillon d'apprentissage.

✔ Le meilleur hyper-paramètre semble être k=3, mais comme nous testons la prédiction sur nos exemples d'apprentissage, cette valeur n'est pas réaliste.

Partie 4 : Evaluation de l'erreur réelle du classifieur appris :

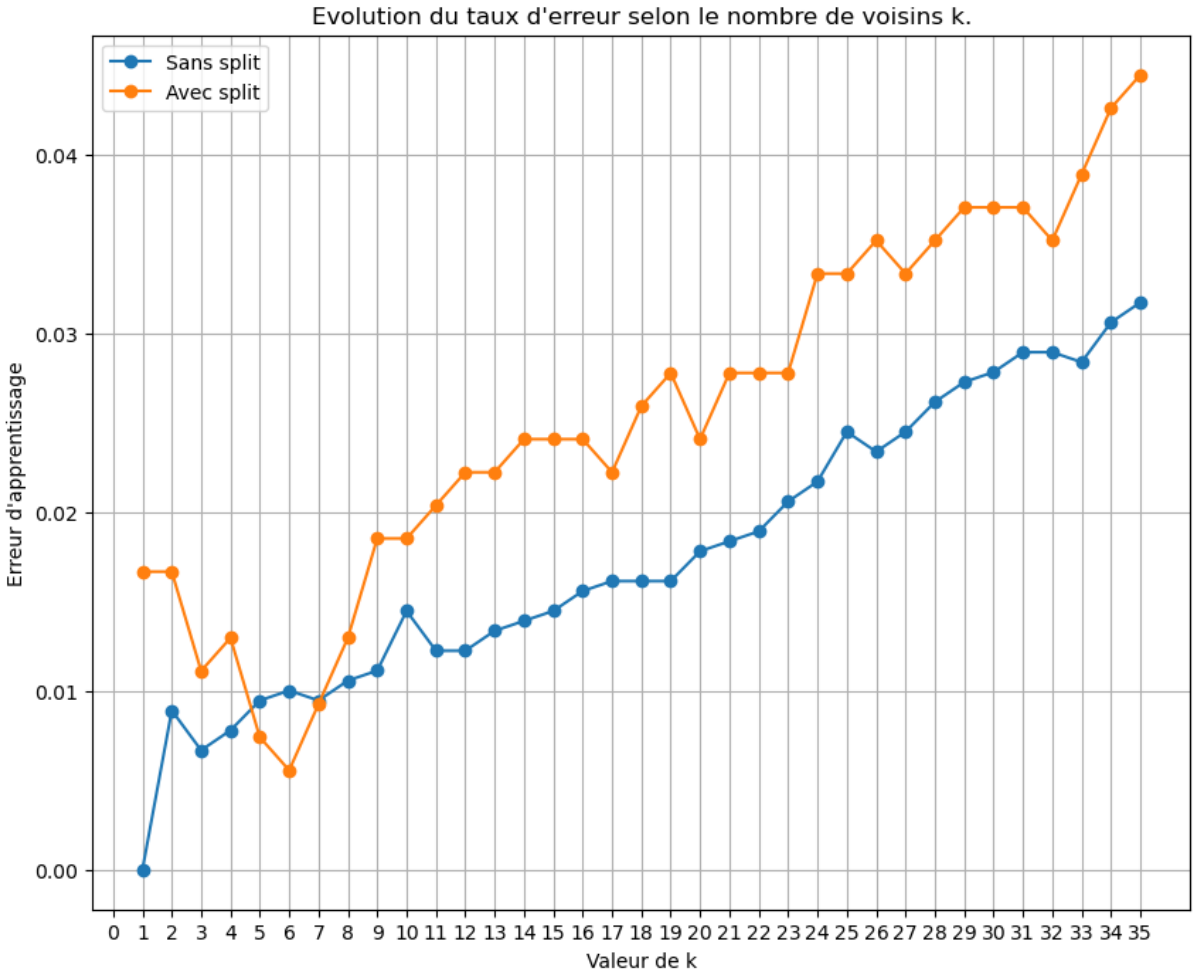
1. Hold-out :

- On remarque que lors de deux appels différents avec random_state = 42, les premiers éléments des tableaux restent les mêmes, contrairement à un troisième appel avec un random_state différent.

- En séparant les données d'entraînement et de tests en deux parties égales, l'erreur d'apprentissage diffère de celle mesurée sans séparation :

    - On obtient une erreur de 1,334% contrairement à une erreur de 0,0067%.

- Variation de k avec un test_split de 0,3 :



✔ L'observation des courbes d'erreur révèle des tendances intéressantes. L'erreur calculée sur l'échantillon de test (en orange) est globalement plus élevée que celle calculée sur l'échantillon d'apprentissage (en bleu), ce qui est attendu. Cependant, l'erreur sur l'échantillon de test fournit une estimation plus fiable de la performance réelle du modèle.

✔ Dans ce cas, le modèle atteint son meilleur score lorsque k est égal à 7, tandis que l'erreur d'apprentissage était plus faible pour k=1. On remarque également que l'erreur n'est plus nulle à un k donné.

✔ On en conclut qu'un modèle qui fonctionne très bien sur les données d'apprentissage (comme dans le cas de k=1) peut ne pas bien généraliser sur de nouvelles données, ce qui est reflété dans l'erreur de test. Le modèle doit éviter l'overfitting.

✔ Le hold-out pratiqué ainsi mène à une estimation de l'erreur réelle qui dépend fortement de l'ensemble train et de l'ensemble test, qui ont été obtenus au hasard, où nous ne contrôlons que leur taille.

 En programmant la répétition de 10 séquences de hold_out, l'erreur réelle d'un kppv avec k = 3 est de 0,01577. Ce qui est sensiblement supérieur par rapport à un hold_out simple.

2. Validation croisée :

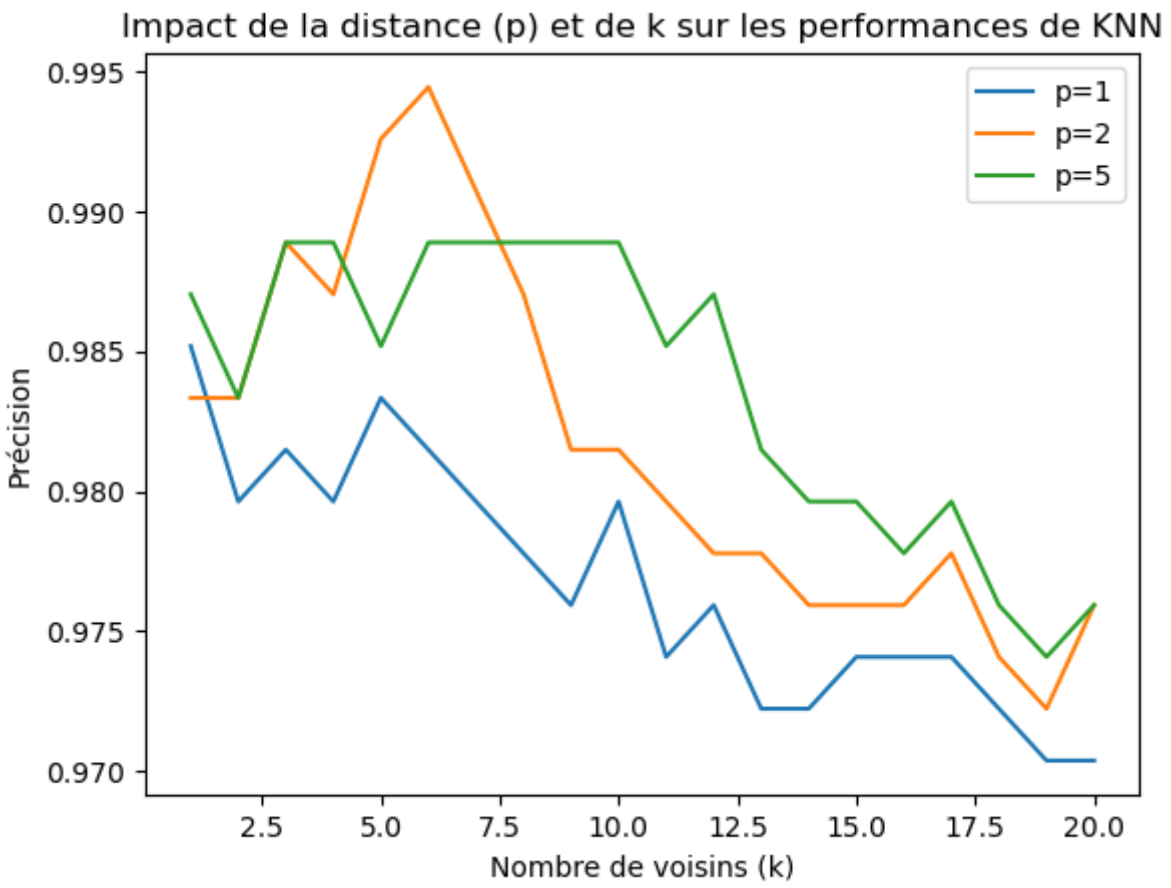
 En répétant l'estimation par validation croisée, on obtient une erreur de 0,02336, ce qui est légèrement supérieur à l'estimation par hold_out répété.

 Lorsque l'on répète l'estimation par validation croisée, on effectue plusieurs itérations de la validation croisée en utilisant différentes divisions de l'ensemble de données en jeux d'apprentissage et de test. Chaque itération peut conduire à des ensembles d'apprentissage et de test différents, et par conséquent, les performances mesurées peuvent varier d'une itération à l'autre. La moyenne des performances sur ces itérations donne une estimation plus stable de la performance du modèle.

 D'autre part, dans l'estimation par hold-out répété, on effectue également plusieurs itérations de séparation de l'ensemble de données en ensembles d'apprentissage et de test, mais il est possible d'obtenir des divisions similaires lors de ces itérations.

Partie 5 : Variations autour de la métrique :

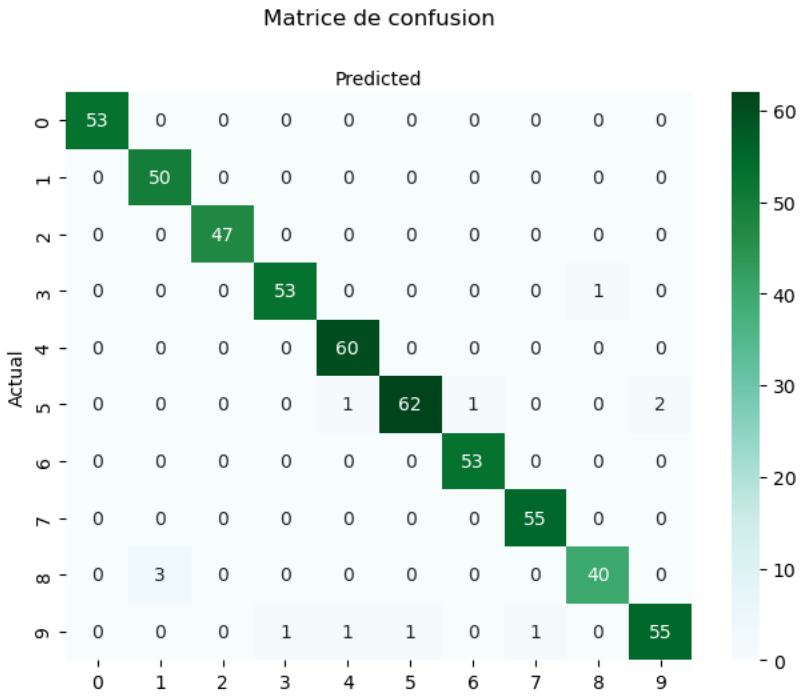
 Impact de la distance (p) et de k sur les performances de Kppv :



- La courbe bleue est pour p=1, l'orange pour p=2 et la verte pour p=5.
- La verte (p=5) semble plus stable dans le cas des données digits avec k<=10, alors que la distance L1 ne donne pas de résultats aussi bons. La distance euclidienne (p=2) donne le meilleur résultat (k=6) mais reste très instable.
- Le choix du meilleur couple d'hyper-paramètres est difficile, ici : soit on prend les meilleurs (k=6, p=2), soit on prend les seconds meilleurs pour privilégier la stabilité.

Partie 6 : Matrice de confusion :

La matrice de confusion obtenue est :



Les confusions observées :

- La classe 3 a été confondue avec la classe 8 une fois (élément [3][8] = 1).
- La classe 5 a été confondue avec la classe 4 une fois (élément [5][4] = 1), avec la classe 7 une fois (élément [5][7] = 1), et avec la classe 9 une fois (élément [5][9] = 1).
- La classe 8 a été confondue avec la classe 2 trois fois (élément [8][2] = 3).
- La classe 9 a été confondue avec la classe 3 une fois (élément [9][3] = 1) et avec la classe 4 une fois (élément [9][4] = 1).

Ces confusions montrent que le classifieur est performant, au regard de l'écriture des chiffres, par exemple le 9 et le 6, qui peuvent être facilement confondues.