
KNN
Algorithm
FROM SCRATCH IN PYTHON

TEST DE PERSONNALITÉ

SOMMAIRE

Partie 1
Base de données, Analyse, Prétraitement et Préparation

1 – Déroulement du programme :

Le fichier ‘Test.py’ est importé dans l’application ‘run.ipynb’.

La maintenance de l’application est effectuée à partir du fichier ‘Test.py’. Les questions y sont enregistrées, les champs d’interactions utilisateur (prompt) récupèrent les données.

Les données sont traitées, un score est établi et une interprétation (A, B, C) est proposée ; le score et l’interprétation sont affichés dans le fichier ‘run.ipynb’.

Les réponses et le traitement sont enregistrés dans un fichier csv.

Les fichiers csv de plusieurs utilisateurs sont ensuite mis à disposition afin de développer le modèle d’intelligence artificielle.

2 – Import des fichiers

Ce premier traitement est effectué dans le fichier ‘import_fichiers.ipynb’.

Les fichiers csv sont importés d’un dossier avec l’aide du module OS de python. Le code sélectionne uniquement les fichiers csv.

Le module Pandas est utilisé ensuite pour concaténer les fichiers importés en un dataframe.

Le dataframe final est ensuite exporté dans un nouveau fichier csv.

3 – Import du dataframe

La suite des traitements est effectuée dans le fichier ‘main.ipynb’.

Le jeu de données présente des valeurs non souhaitées.

4 - Analyse

Deux options sont possibles à ce stade, au vu du nombre important de Nan.

Il reste 41 lignes sur les 225 (18%) de données disponibles en cas de suppression.

Ce qui veut par ailleurs dire que 82 % des données seront tronquées en cas de stratégie de remplacement des Nan par la valeur la plus présente.

Aucune des deux solutions n’est réellement satisfaisante.
D’autant plus que sur ces données, l’interprétation A n’est présente qu’une seule fois.
L’apprentissage ne sera sans doute pas concluant.
Néanmoins.

Après test avec Sklearn, la stratégie de remplacement donne un score de 71 %, la stratégie de ne conserver que les 41 lignes ‘correctes’ donne un score de 77 %.

Le choix est fait de ne conserver que les données correctes, les NaN sont donc supprimés.

Nous avons deux targets possibles :
- le score
- l’interprétation

Il est demandé d’effectuer une classification, aussi c’est l’interprétation qui est choisie, le score est écarté.
Le score impliquerait une régression.

Partie 2 : Développement et entraînement d’un modèle KNN

1 - Modèle from scratch

Trois fonctions de calcul de distances sont initialisées, la distance euclidienne, la distance Manhattan, la distance Minkowski.

Pour ces trois fonctions, le module Numpy est utilisé afin d’utiliser le calcul matriciel, pour alléger l’écriture.

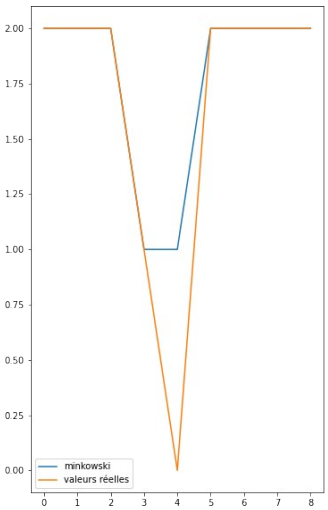
La fonction distance retourne les distances sur le Dataframe.

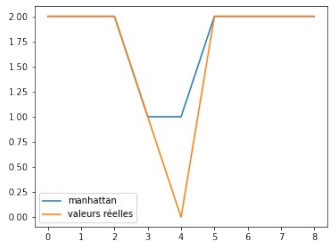
La fonction predict calcule la distance minimum et prédit l’interprétation.

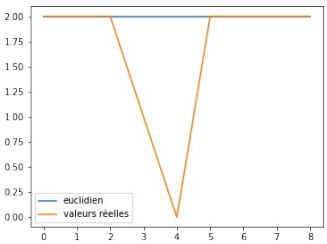
Voici les prédictions selon les différentes fonctions :

- minkoswski : 0.88
- manhattan : 0.88
- euclidien : 0.77

Ci-dessous les graphiques pour les différentes fonctions :







2 – modèle avec Sklearn

Après différents essais, le modèle créé a une précision meilleure avec les paramètres suivants :

- distance : manhattan (p=1)
- k : 4

L’accuracy est de 80 % sur les données d’entrainement, 77 % sur les données de test.

Le modèle Sklearn sera plus fiable que le modèle from scratch, puisque il inclut des tests sur 5 fois les données d’entrainement.

C’est ce modèle qui est enregistré via Joblib.

Partie 3 : Mettre en place la solution dans l’application de test de personnalité

L’étape de traitement des données ainsi que le modèle d’intelligence artificielle sont implémentés dans le fichier ‘Test.py’.

Les résultats sont affichés lorsque les données utilisateurs sont traitées.

Conclusion :

L’étape suivante, hors-projet, est d’enregistrer les erreurs de l’intelligence artificielle par rapport aux vraies interprétations afin d’affiner notre modèle.