DM

Regarder pour comprendre l'enjeux de cette partie

https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning
Le site logiciel du logiciel Scikit-learn est : http://scikit-learn.org/stable/index.html La documentation en ligne est complète, et devra être consultée chaque fois que nécessaire : http://scikit-learn.org/stable/documentation.html.
Des tutoriaux sont disponibles à l'adresse suivante : http://scikit-learn.org/stable/tutorial/index.html

Protocole :

1. Implémenter algorithme les k-plus proches voisins

Utiliser la distance euclidienne pour cet algorithme. Utiliser le fichier d*ata*_eleve_GPS_Groupe.csv
Tester votre code sur les coordonnées de fanny avec k variant de 1 à N
Expliquer les résultats

2. *Création et entrainement d'un classifieur*

Utiliser 20% de donner test et 80% pour l'apprentissage soit environ (8 et 20)

3. *Evaluation de l'erreur du classifieur appris*

Pour chaque valeur de k faire l'évaluation de l'erreur du classifier. Présenter le résultat sous forme de tableau

4. Sélection de modèle. Donner le k choisi (le modèle) Sélectionner et donner le k pour lequel l'erreur est la plus faible et stable.

Le fichier **data_eleve_GPS_Groupe.csv** contient les informations suivantes : nom_eleve, latitude, longitude, station. Les élèves sont répartis en groupes. Le nom du groupe correspond au nom de la station la plus proche du lieu d'habitation de l'élève qui est définie ici avec les coordonnées GPS (latitude, longitude). Dans cette partie on utilisera ***uniquement les colonnes*** suivantes : latitude, longitude, station.

Visualiser les données
Afficher les données sous la forme de nuages de points. Le graphique doit avoir comme titre « cartographie » Les axes du graphique doivent être nommés (longitude, latitude). On utilisera la longitude et la latitude comme respectivement l'abscisse et l'ordonné du graphique. Ajouter une légende avec un code couleur pour les différentes stations.

5. Analyse de data

- Le fichier **echantilon_eleve.csv** contient 3 élèves dont on veut connaitre le groupe. Ajouter au graphique cartographie 3 points supplémentaires avec une mise en forme de couleur noir et symbole +. Cette mise en forme ne doit être utilisée que pour ces points du graphique. Modifier la légende en ajoutant l'étiquette « inconnue » Ajouter un cercle en noir permettant de voir autour de la cible les voisins les plus proches.
- Déterminer l'appartenance de chaque élève à l'aide de l'algorithme k-plus proches voisins (kppv). Modifier le graphique en conséquence.
- Ecrire la fonction, Insert_echantilon(nom,longitude,latitude,groupe,data_eleve_GPS_Groupe.csv) qui ajoute en fin du fichier **data_eleve_GPS_Groupe.csv** les nouvelles données.

- Ecrire la fonction insert_distance_station(data) qui ajoute une colonne « distance » en header et ajoute les distances correspondantes entre élève et station. Faire un appel de la fonction distance_entre_deux_Points_GPS **NE PAS ARRONDIR !**
  Par exemple :

	A	B	C	D	E	F
1	nom_eleve		***longitude***	***latitude***	station	distance en m
2	Fanny	1	2,315733	48,84686	DUROC	92,5389232860018

- Ecrire la fonction ordre_par_groupe(**argu_a_choisir**)
  La fonction classe les données par ordre alphabétique en fonction des noms de groupes, c'est-à-dire on attend ici DUROC / Vaneau/ Sevres
- Ecrire la fonction **ordre_fusion_distance_groupe(argu_a_choisir)**
  Ecrire une fonction tri par fusion de manière récursive qui pour chaque groupe classe par ordre croissant les distances (sans perte de données)
- Ecrire un algorithme qui retourne les résultats précédant dans **data_eleve_GPS.csv** et qui arrondi les valeurs la colonne distance au mètre supérieur.

Itinéraires de métro

La recherche d'itinéraires repose sur un calcul du plus court chemin dans un graphe value (c'est-à-dire un graphe ou chaque transition est munie d'une valeur qui correspond à la longueur ou durée de la transition).
À l'aide des coordonnées GPS de différentes stations construire le graphe associé à la carte.
Le graphe est considéré dans un premier temps comme non orienté et non pondérée.
Écrire la matrice adjacente associée à ce graphe.
Pour des raisons de simplification nous utiliserons uniquement que les lignes suivantes 1,4,8,10,12,13.
Et uniquement les noms des stations dans le fichier (certaines stations sont considérées comme fermées)
A partir des lignes de Metro, construire un graphe dont les sommets sont les stations et ou les arêtes correspondent à des étapes des lignes. On supposera que la durée d'une étape est 1min30. Appeler l'algorithme de Dijkstra sur ce graphe pour en déduire des itinéraires et des temps de parcours minimaux entre deux stations. On affichera aussi l'itinéraire à suivre.
Implémenter algorithme de Dijkstra en python de la manière qui suit :

- À l'aide d'une classe (option)
- **Sous forme explicite et itérative *(obligatoire)**
- **A l'aide d'une matrice**
- Forme récursive (option)

N'utiliser que des bibliothèques dit de base.
Par exemple tester votre code pour (départ =Duroc, arrivé= Louvre)

6. Pondération

A partir des lignes de Metro, construire un graphe dont les sommets sont les stations et les arêtes correspondent distance entre deux stations.

A l'aide de la fonction **distance_entre_deux_Points_GPS** (cf fichier)

Calculer la distance entre chaque station pour toutes les lignes. **(Arrondir au m supérieur)**
Modifier la matrice adjacente.

Optimisation :
Ecrire une fonction **Creation_matrice(#argument à choisir**) que génère automatique la matrice précédente.
Par exemple tester votre code pour (départ =Duroc, arrivé= Louvre)

- Statistiques sur le réseau

À partir des algorithmes de la question précédente, on ajoutera le calcul de plusieurs mesures sur le réseau :

Optimisation :

Nombre minimal de correspondances permettant de se rendre partout dans le réseau depuis n'importe quelle station.
Nombre minimal de correspondances permettant de se rendre partout dans le réseau depuis n'importe quelle station on attend minimal.
Les stations les plus éloignées dans le réseau.

7. Base de données

Modèle de conception
Définir Type d'Association, Type d'Entité.
Un groupe est composé de plusieurs élèves. Un élève appartient à un seul groupe.
Une ligne de métro est composée de plusieurs stations.
Un élève effectue un trajet en partant d'une station de départ jusqu'à un point d'arrivé.
Un trajet est composé d'un ensemble de stations.
Un élève est défini par son prénom et son adresse GPS.
Une station est définie par son nom, une adresse et des coordonnées GPS.
Définir le Modèle de conception.

8. ***Base de données*** utiliser **data_eleve_GPS.csv**

Les commandes SQL doivent être écrites dans un fichier.txt
Utiliser les fichiers data pour compléter la base
Cette partie est indépendante de la question précédente
Id correspond à Clé primaire + auto-incrément
Définir une base de données nommée MBDD_votre_Prenom
Créer une table élève (**id_eleve**,prenom,longitude,latitude)
Créer une table groupe (**NOM, id_eleve, NOM_station,distance)**
La distance entre élève et la station la plus proche. En mètre.
Créer une table station (**NOM_station**, longitude, latitude)
Créer une table trajet (**à vous de trouver les attributs**)

9. ***PHP javascript html et css***

Définir un site web avec plusieurs pages accessibles avec des onglets nommées par Page 1, Page 2 , Page 3.
Page 1 est composé de :
Un champ de saisie (pour les coordonnées GPS)
Un menu déroulant avec la liste des stations.
Un bouton nommé (rechercher la destination)

DM

Principe :
Si l'on clique sur le bouton (rechercher la destination)
La page web retourne d'une part le chemin le plus court, distance totale effectuée.
Sur une carte (graph) tous les chemins doivent être représenter en noir seul le chemin le plus court est en rouge.

Page 2 est composé de :
Un champ de saisie (pour les coordonnées GPS)
Un champ de saisie (rayon en mètre)
Un bouton nommé (rechercher nombre de voisin)
Principe :
Si l'on clique sur le bouton (rechercher nombre de voisin)
La page web retourne sur la carte le nombre d'élèves autour du point en rouge des coordonnées GPS entrées par l'utilisateur. La page web afficher un cercle en rouge centre au point rouge de rayon entrées par l'utilisateur. Sur la légende de la carte doit apparaitre ne nombre total d'élèves présent et le « pourcentage de groupe » par exemple dans 500m : 6 élèves avec gr : 50% sont DUROC 50% Vanneau.
Reprendre le code couleur pour les groupes
Page 3 est composé de :
Un champ de saisie (pour les coordonnées GPS)
Un champ de saisie (rayon en mètre)
Un bouton nommé (rechercher le voisin le plus proche)
Principe :
Si l'on clique sur le bouton (rechercher le voisin le plus proche)
La page web retourne sur la carte le prénom de l'élèves, la distance entre les coordonnées GPS entrées par l'utilisateur.
Schéma attendu