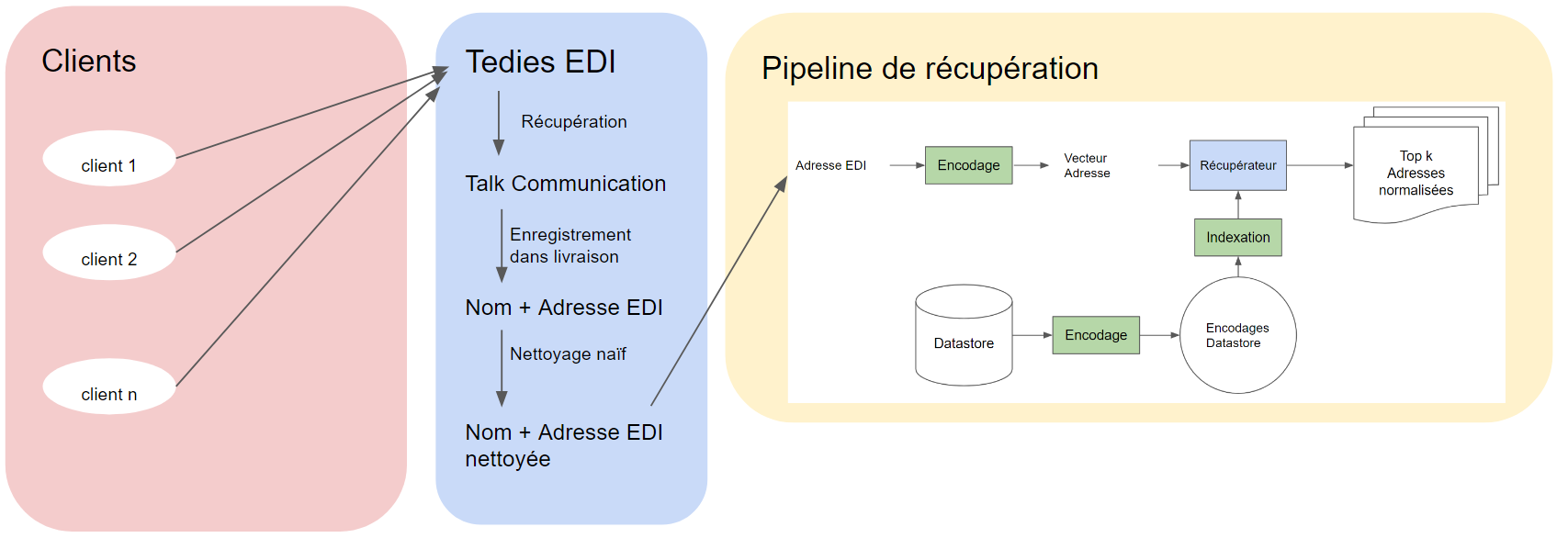
1. Objectifs du document

La conception d’un workflow de test et de comparaison réel pour la solution IA de Moundir sur l’identification des destinataires. La complexité réside dans l'intégration du modèle IA de récupération d’adresses similaires aux adresses reçues via l’EDI en temps réel et par conséquent le ou les destinataires associés et enfin effectuer des comparaisons entre les destinataires trouvés par l’algorithme “destinataire” de Stéphane. Le but ici est de pouvoir mesurer le pourcentage des destinataires correctement retournés de la base d'Upsilon (table des destinataires) en première position. Cela permettrait par la suite de valider ou non la solution IA comme solution complémentaire à l’algorithme de “destinataire”.

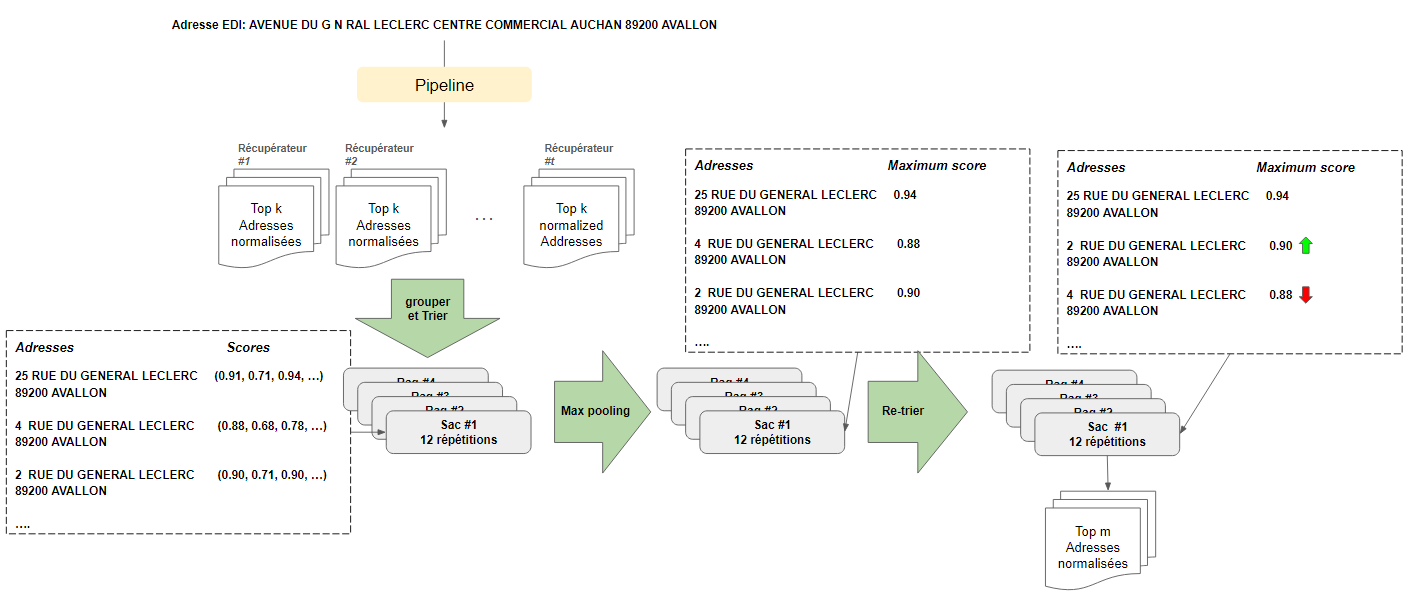
1. Documentations connexes
2. Critères de validation
   1. Minimums
      1. Minimums Pré-Installés

* Création d’un datastore d’adresses postales normées:
  + Chacune des adresses est associée à la ou les clés destinataires correspondantes.
  + Ces adresses seront construites de la table “destinataire” via les clé étrangères la reliant aux tables “voie”, “typevoie” et “commune”
* Pré-enregistrement des différents encodages vecteur du datastore créé:
  + Ce sont des vecteurs qui sont supposés stocker l’information de similarité sémantique des adresses de destinataires.
  + Ils seront utilisés par la suite comme des bases de référence parmi lesquelles nous allons récupérer les adresses normées les plus similaires.
* Choisir et mettre en place le récupérateur d’adresses similaires:
  + C’est généralement un algorithme des plus proches voisins (kNN).
  + Ce dernier devra être choisi selon sa capacité à récupérer rapidement un nombre défini d’adresses voisines à l’adresse EDI.
* Pré-enregistrer l’arbre de recherche des adresses voisines de chaque type d'encodage du datastore via l’algorithme kNN. Le pré-enregistrement devra nous éviter de calculer l’arbre à chaque fois qu’une adresse est reçue.
* Mettre au point le pipeline qui permet :
  + de prendre une adresse EDI, l’encoder en différents types de vecteurs,
  + les passer par le kNN qui s'en chargera de récupérer les k premiers indices des vecteurs d’adresses similaires pour chaque type.
* Mettre en place le processus de vote qui s'en chargera de:
  + tirer une liste finale d’adresses voisines parmi les différentes listes de k voisins des types d’encodage.
    1. Minimums au temps réel
* Utiliser les fonctions déjà existantes pour la récupération des noms et adresses postales des livraisons via l’EDI du jour.
* Effectuer un nettoyage naïf et rapide des champs Adresse1 et Adresse2. Le but sera d’enlever quelques symboles parasitaires de l’adresse EDI.
* Trouver le destinataire associé à travers la comparaison entre le nom associé à l’adresse EDI et les noms associés aux adresses normées récupérées.

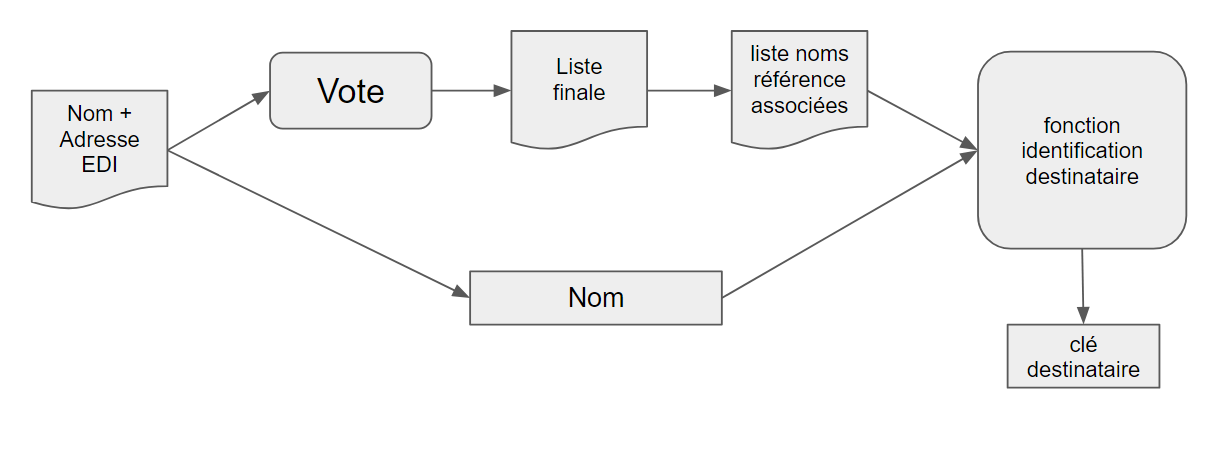
1. Processus général de la solution IA
   1. Pipeline de récupération



* 1. Création de liste finale



* 1. Trouver le nom destinataire



1. Étapes du vote pour la construction de la liste finale

Le principe de vote repose sur deux types de valeurs:

* Le nombre de répétitions d’une adresse normée du datastore parmi les différentes listes des récupérateurs.
* Les scores de similarité associés à l’adresse répétée (nombre de scores = nombre de récupérateurs ayant sortie l’adresse)

Et par conséquent, les étapes qu’on devrait suivre pour construire notre liste finale sont les suivants:

* Groupement des adresses normées dans des sacs de la façon suivante:
  + comptage du nombre de répétitions de chaque adresse.
  + mettre dans le même sac les adresses avec le même nombre de répétition
  + tri des sacs selon la valeur maximale des nombres de répétitions.
* Max pooling des scores de similarité de chaque adresse dans les sacs:
  + récupération des scores de similarités de chaque adresse.
  + choisir le maximum de ces scores et le garder pour l’adresse.
* Re-triage des adresses dans les sacs:
  + dans chaque sac, nous trierons les adresses selon le score maximal.
* Concaténation des sacs et tirage de la liste finale:
  + définir le nombre “m” de voisins similaires voulu.
  + sortir les m adresses normées les mieux classées.