recordlinkage 0.16

RecordLinkage recordlinkage.readthedocs.io	**Librairie Python pour le matching**	
	Système d'exploitation :	Multi-plateforme
	Développé par :	Jonathan de Bruin
Open source (BSD)	Personne de contact :	Vandy.Berten@Smals.be

Fonctionnalités

Recordlinkage est une librairie Python permettant le « matching » (lier des enregistrements entre deux tables Pandas, librairie de manipulation et d'analyse de données) ou le « dédoublonnage » (détecter des doublons au sein d'une table Pandas). L'utilisation de la librairie se fera en général en trois temps :

1. On commence par définir le « blocking », soit la méthode qui va déterminer quelle ligne d'une table sera comparée à quelle autre. On peut soit faire une comparaison complète (chaque ligne avec chaque ligne), soit faire une comparaison par bloc (on ne compare deux lignes que si la valeur d'une colonne définie est la même ; par exemple le code postal), soit par « voisinage » (on ne compare deux lignes que si leur valeur n'est pas plus éloignée qu'un paramètre, dans l'ordre alphabétique).
2. Pour chaque couple de lignes sélectionné à l'étape précédente, on définit un certain nombre de « *features* », à savoir une méthode et des paramètres pour déterminer si la comparaison est valide ou non. Par exemple, 1 si le champ « gender » est le même de chaque côté, 0 sinon. Ou 1 si la distance de Levenshtein entre les prénoms est inférieure à 2, 0 sinon.
3. On peut ensuite déterminer, pour chaque couple comparé et en fonction des « *features* », ceux qui sont valides et ceux qui ne le sont pas. Cette classification peut soit se faire de façon déterministe, en précisant explicitement les combinaisons valides, soit en utilisant une des nombreuses méthodes de *machine learning* proposée dans la librairie.

Conclusions & Recommandations

Recordlinkage est une librairie efficace, populaire et puissante. Elle ne remplace pas une suite complète de gestion de Data Quality, mais permet déjà d'obtenir des résultats tout à fait satisfaisants dans de nombreuses situations auxquelles est confronté un data scientist.

Tests & Résultats

```
(…)
# Prepare blocking
indexer = recordlinkage.Index()
indexer.block('zipcode')
candidate = indexer.index(df_a, df_b)
# Add comparer
c = recordlinkage.Compare()
c.exact('name_a', 'name_b',
        label='NAME')
c.string('name_a', 'name_b',
         method='jarowinkler',
         threshold=0.9,
         label='NAME_jaro')
c.exact('street_a ', 'street_b',
        label='STREET')
c.string('street_a', 'street_b',
         method='levenshtein',
         threshold=0.7,
         label='STREET_lev')
# Compute features
feat = c.compute(candidate,
                 df_a, df_b)
# Use features
valid = (feat.NAME & feat.STREET_lev)
     | (feat.NAME_jaro & feat.STREET)
```

La librairie recordlinkage offre toute une série d'outils permettant de réaliser toutes les étapes nécessaires au matching (illustré ci-contre entre les deux dataframes Pandas df_a et df_b) ou au dédoublonnage.

En premier lieu, plusieurs méthodes permettent le nettoyage : suppression d'accents, de ponctuation, standardisation de numéros de téléphone…

Vient ensuite une série d'algorithmes phonétiques, permettant typiquement de générer des « blocs », au sein desquels des comparaisons détaillées seront faites : Soundex, Metaphone, NYSIIS…

On peut maintenant déterminer les comparaisons qui seront effectuées (« prepare blocking » dans le code joint) : «full » pour comparer chaque ligne de df_a avec chaque ligne de df_b ; « block » pour ne comparer qu'au sein de blocs définis à l'étape précédente (dans l'exemple, on exige un 'zipcode' identique) ; « sortedneighbourhood » pour également comparer des blocs « proches ». Il est possible de combiner plusieurs méthodes, pour augmenter le nombre de tuples traités.

On arrive enfin au « cœur » du métier, en définissant les « features » (« Add comparer »). Dans l'exemple, on va créer une *feature* « NAME » valant 1 si le champ df_a.name_a est exactement égal à df_b.name_b (et 0 sinon), ou « NAME_jaro », si la similarité de Jaro-Winkler est supérieure à 0.9. Il est également possible d'utiliser des comparaisons numériques, géographiques, de dates (tolérant par exemple les inversions mois-jour, erreur très fréquente), ou même n'importe quelle méthode définie par le développeur.

Il ne reste plus qu'à utiliser les résultats : on considère dans l'exemple qu'une comparaison est valide uniquement si le nom est exactement égal et la rue proche, ou l'inverse (et le code postal identique, puisqu'il s'agissait de la clé de « blocking »).

Recordlinkage offre de la classification non-supervisée (sépare automatiquement les candidats valides des autres) ou supervisée (sur base de données labellisées). Nous n'avons pas été convaincus par le résultat de la version non-supervisée, mais c'est à évaluer au cas par cas.

Il s'agit donc d'une librairie complète, performante (supporte le multithreading et basée sur Pandas), personnalisable, que nous avons déjà pu utiliser avec satisfaction lors de multiples projets.

Conditions d'utilisation & Budget

recordlinkage est une librairie Open-Source gratuite, disponible sur PIP ou Conda.