Documentation technique

Algorithme d'estimation données ECLN

ADEQUATION 2018

INFORMATIONS

Nom du projet : Algorithme d'estimation des données ECLN

Type de document : Documentation technique

Date : 13/08/2018

Mots clés : ECLN – Algorithme – Estimation

TABLE DES MATIERES

1) Résumé du document :

Ce document est la documentation technique officielle du projet Algorithme d'estimation de données ECLN. Il est divisé en quatre parties :

- Rappel de l'objectif de l'algorithme.
- Définition des règles d'estimation.
- Documentation technique du code : choix des technologies et déroulement de l'exécution du script, optimisations possibles et fonctions à implémenter.
- Utilisation de l'application.

2) Rappel de l'objectif de l'algorithme et présentation des données :

Pour réaliser ses études, Adéquation travail avec des données issus des enquêtes sur la commercialisation des logements neufs (ECLN). Mais certaines données sont soumises au secret statistique. Le but de cet algorithme est de calculer si possible ces données sinon les estimer. L'algorithme se base sur les données ECLN au niveau des communes et au niveau des EPCI. Une ligne au niveau EPCI, pour un code siren donné, représente la somme des volumes au niveau commune qui ont le même code siren.

Voici comment les données sont présentées sous Excel au niveau communes :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
REGION	DEP	NOM DEP	New insee 2016	VILLE INSEE	siren2017	Coll MEV	Coll Réservations	Coll Annulés	Coll Chgt dest	Coll Encours	Coll Prix Réservations	Ind MEV	Ind Réservations	Ind Annulés	Ind Chgt dest	Ind Encours	Ind Prix Réservations
LORRAINE	57	MOSELLE	57055	BAZONCOURT	200067957												
LORRAINE	57	MOSELLE	57145	COINCY	200067957												

En ligne nous avons les communes. En colonne nous avons le nom des régions, le numéro et le nom des départements, les noms et les codes INSEE. Le code Insee est un code numérique élaboré par l'Institut national de la statistique et des études économiques (Insee), concernant les collectivités, la géographie, les populations et les entreprises. Et enfin les codes siren créé par l'INSEE qui sont des identifiants au niveau de l'EPCI. Ce code indique à quel EPCI chaque commune appartient.

Ici nous avons un exemple concernant l'année 2005. Sur cette année, pour les logements collectifs (Coll) et individuels (Ind) nous avons plusieurs données :

- Le volume des mises en vente de nouveaux logements proposés à la commercialisation durant l'année n (MEV). C'est la seule donnée qui ne soit pas soumise au secret statistique. Cette donnée est obligatoirement communiquée.
- Puis le volume de réservation de logement.
- Puis le volume d'annulation d'achat de logement (réservations annulées).
- Ensuite le volume de changement de destination. Lorsque certains logements ne parviennent pas à être vendus, certains logements changent de programme. C'est un lot de logement qui étaient destiné à être du logement libre mais qui finit par être vendu en tant que logement social ou inversement.
- Ensuite l'encours qui représente le volume de logements proposés à la vente encore non réservés à l'année n.
- Enfin les prix des réservations qui sont des prix moyens à la réservation sur la commune.

Pour chaque ligne/commune, nous avons ces colonnes pour les années allant de 2005 à 2017.

Les données soumises au secret statistique à estimer sont celles qui sont soit manquantes soit marquées par 'nd' (données non communiquées) dans le fichier Excel.

3) Règles de calcul et d'estimation

3.1) Liste des règles

Les règles de calculs sont implémentées de manières à ce que les données après traitement soient les plus fiables possible. C'est-à-dire que les calculs sont effectués en premiers puis les estimations en derniers. Une priorité est définie pour les règles à implémenter en fonction de leurs fiabilités.

Voici les règles mise en place par ordre de priorité :

- (0) S'il n'y a aucune activité pour une commune donnée sur l'ensemble des années (2005 à 2017), alors remplir avec des 0.
- (1) Les données de mise en vente étant non soumises au secret statistique, s'il subsiste des cellules vide dans le fichier, on les rempli avec un 0.
- (2) On peut calculer l'encours de l'année n-1 si les données de l'année n sont complète. L'encours en n-1 se calcul de la manière suivante :

$$EC(n-1) = EC(n) - MEV(n) + R(n) - A(n) - CHD(n)$$

- (3) Si, dans un regroupement de commune par code siren, une seule ligne présente des données manquantes, on peut la déterminer grâce à la ligne au niveau EPCI ayant le même code siren.
- (4) Une ligne au niveau EPCI étant la somme des volumes au niveau des communes ayant le même code siren, si on a un 0 dans une des colonnes au niveau des EPCI, cela signifie que le volume total au niveau des communes pour la même colonne est nul. On peut donc mettre des 0 dans les colonnes correspondantes au niveau des communes si on a des 0 au niveau des EPCI.
- (5) La règle qui va suivre est cette fois ci une règle d'estimation et non un calcul. Après avoir appliqué toutes les règles précédentes, on applique la règle de trois. Cette règle permet de faire des estimations sur les colonnes Réservations, Annulations et Changement de destination en fonction du volume de mise en vente.
- (6) On recalcul la colonne Encours avec les données estimées grâce à la règle de trois avec la formule suivante :

$$EC(n) = EC(n-1) + MEV(n) - R(n) + A(n) + CHD(n)$$
$$\text{Ou} \quad EC(n-1) = EC(n) - MEV(n) + R(n) - A(n) - CHD(n)$$

3.2) Implémentation

Les estimations se font de la manière suivantes. Les premières fonctions (1, 2 et 3) s'effectuent directement sur le dataframe original. Pour certaines des fonctions implémentées (3,4 et 5), on extrait les données à traiter en sous-groupe (par code siren)

pour pouvoir faire des estimations et on les ‘replace’ ensuite dans le dataframe original en faisant une mise à jour (avec la fonction pandas : **update()**). La dernière fonction se fait directement sur le dataframe original.

4) Documentation technique du code

4.1) Technologies utilisées

Langage utilisé :

- Python 3.6.5

Librairies utilisées :

- Pandas 0.22.0

4.2) Exécution de l'algorithme

1- importation des librairies (L 1-11).

2- ouverture des fichiers qui contiennent les données sélectionnées pour l'apprentissage (L 17-20).

3- sélection des colonnes dans les 2 dataframes (L 24-25).

4- fonctions de nettoyage et sélection des données à traiter. Les données à traitées sont stockées dans '**dataframe_commune_valide**'. (Attention les données qui ne sont pas à traiter sont stockées dans les dataframe '**data_commune_zero**' et '**data_commune_reste**'. Il faudra les concaténer verticalement avec le dataframe principal une fois les données estimées). Ce choix de découpage a été fait pour réduire le nombre de données à parcourir lors des traitements. (L 30-102).

5- fonctions de traitement des données (L 106-219).

6- Dans les colonnes "MEV" (mise en vente, non soumis au secret statistique), s'il y a un vide, on met des 0 (L 222-230).

7- Calcul de l'encours à l'année n-1 si les données de l'année n ne sont pas manquantes (L-233-250).

8- Pour un code siren donné, si au niveau EPCI il y a un 0 dans une colonne, on reporte ce 0 au niveau commune dans toute la colonne correspondante (L 253- 278).

9- Calcul de la dernière ligne si dans un regroupement de communes au niveau par code siren, une seule ligne est manquante (L 282-320).

10- Estimation par règle de trois (L 324-343).

11- (en développement) Calcul des Encours avec les données estimées avec la règle de trois (L 347-431*). *le code n'est pas terminé.

4.3) Optimisations

- Certaines partie de code de certaines fonctions sont relativement rigide aux changements de structure du fichier ECLN. Si la structure change, certaines fonctions seraient à adapter en conséquent. Par exemple, l'estimation de l'encours à l'année n-1 requiert d'avoir accès aux données à l'année n+1 (cf. formule). L'accession à ces données se fait en comptant le nombre de colonnes qui sépare la cellule à calculer de l'encours n-1 et les données de l'année n.

- Optimisations au niveau du temps d'exécution qui actuellement est d'environ 30 minutes. Les fonctions de mise à jour dans le dataframes occupent la majeure partie de ce temps (60%).

- Optimisation au niveau des fonctions utilisés pour accéder aux données. Cela pourrait avoir un impact sur le temps d'exécution également puisque celles-ci se trouves, la plupart du temps, dans des boucles. La librairie pandas étant très vaste, il existe plusieurs moyens de réaliser une tâche. Certaines sont plus rapide que d'autres et il y a possibilité de gagner du temps sur certaines de ces fonctions.

- Utilisation de dataframe multi index plutôt qu'un dataframe avec un seul index. Cela éviterait d'avoir des noms de colonnes trop longs et de rendre la mise en forme des données et l'accession aux données plus claires. Cependant cela impliquerait de refaire toutes les accessions aux données pour être adaptées à un dataframe multi index.

- Il est aussi possible de fusionner certaines des règles d'estimation pour réduire le nombre de parcours de tout le dataframe. Les fonctions (3) et (5) peuvent être regroupées en une seule.

4.4) Fonctions restantes à implémenter

Premièrement, il faut intégrer les données ECLN 2017 aux données actuellement utilisées par l'algorithme qui concerne les années 2005 à 2016.

Ensuite, il reste une fonction à implémenter. A la suite des estimations de la règle de trois, il faut recalculer les encours. Il faut utiliser les formules suivantes :

- (1) $EC(n) = EC(n-1) + MEV(n) - R(n) + A(n) + CHD(n)$
- (2) $EC(n-1) = EC(n) - MEV(n) + R(n) - A(n) - CHD(n)$

Problème : Sachant que nous avons déjà estimé les colonnes de réservations, d'annulations et de changements de destination, pour pouvoir utiliser ces formules, il nous faut au moins l'encours à l'année n ou l'encours à l'année n-1. Or aucunes des colonnes encours n'ont été estimées. Cependant, il est possible qu'il y ait une variable 'encours' qui soit valide soit parce

qu'elle était déjà présente dans le fichier, soit parce qu'elle a été calculée grâce à la fonction (1).

Stratégie :

- Il faut d'abord vérifier s'il y a déjà un encours sur une ligne/commune. (Une fonction est déjà implémentée pour cette partie **cherche_encours_commune()**)
- Si oui, on calcule les encours de toutes les années avec les formules. On détermine celles des années précédentes avec la formule (2) et celles des années suivantes avec la formules (1). (Une fonction est déjà implémentée pour cette partie mais il reste à la tester **calcul_encours_commune()**).
- Sinon, il faut effectuer une règle de trois sur une colonne Encours. Puis à partir de celle-ci, utiliser une des 2 formules données précédemment pour calculer le reste des colonnes encours.
- En principe, on part de l'année 2006, donc il faudrait appliquer la règle de trois sur la colonne Encours de cette année puis calculer l'encours des années suivantes avec la formule.
- Cependant, il se peut qu'il n'y ait pas d'activité/pas de volume au cours de cette année 2006. Donc il faudra vérifier pour chaque année s'il y a du volume pour l'encours.
- Dès qu'une colonne Encours avec du volume a été trouvée, on applique la règle de trois sur cette colonne puis on calcule les encours des autres années avec une des 2 formules.

Rappel de la règle de trois. Les estimations dans une colonne se fait proportionnellement aux volumes totaux de mise en ventes des communes que l'on retrouve au niveau EPCI (une ligne au niveau EPCI représente la somme des volumes au niveau commune pour un même code siren. On peut avoir plusieurs communes pour un EPCI). Par exemple on a une ligne EPCI avec un volume total de 20 mises en vente et un volume de 10 en encours réparties sur 2 communes. La première présente 12 mises en vente et la seconde 8. La première commune 'possède' 60% soit ($\frac{12}{20}$) des parts de mise en vente et la seconde 40% soit ($\frac{8}{20}$). Pour déterminer, les encours de ces colonnes, on répartit le volume total d'encours au niveau EPCI de 10 sur les lignes des communes. La première commune aura donc un volume de 6 en encours ($\frac{12}{20} * 10$) et la seconde 4 ($\frac{8}{20} * 10$). Le code concernant cette fonction se situe aux lignes 348 à 431.

Apres l'implémentation de cette fonction, il faut rapatrier les données que qu'il ne fallait pas traiter au dataframe original. Il faut donc concaténer les dataframe '**data_commune_zero**' et '**data_commune_reste**' à '**dataframe_commune_valide'**.

Enfin, il faut stocker le dataframe dans un fichier csv avec la fonction suivante :
dataframe.to_csv(file_name, sep='\t', encoding='utf-8').

Le dataframe enregistré conservera peut-être l'index numérique qui lui a été donné dans l'algorithme. Pour le retirer, il faudra rajouter le paramètre (index=False) dans la fonction précédente ou tout simplement le retirer via Excel.

5) Utilisation de l'algorithme

Dans LibreOffice Calc (équivalent Excel), remplacer toutes les virgules par des points et retirer tous les symboles '€'. Retirer également les cellules contenant '#VALEUR'.

Ensuite Mettre dans des fichiers csv distinct les feuilles 'niveau epci' et 'niveau commune' avec LibreOffice (pour avoir des fichiers csv formaté correctement).

Renseigner dans l'algorithme les chemins vers les fichiers csv.

Exécuter l'algorithme en ligne de commande.