

1 Parseig de dades

L'script que realitza el parseig de les dades és: **transform_to_arff.py**

1.1 Línia de comandes

Per parsejar les dades passades per la línia de comandes, s'ha utilitzat el mòdul de python **argparser**. Aquest ens permet posar arguments opcionals per a ser utilitzats en el parseig. Per tant els diferents arguments que es poden passar són:

- **dataset**: argument en el qual s'indica la ruta cap al fitxer .csv que conté les dades a parsejar.
- **train**: s'indica el nom a posar pel fitxer de train incloent l'extensió .arff, és a dir, <nom-fitxer>.arff.
- **test**: s'indica el nom a posar pel fitxer de test incloent l'extensió .arff, és a dir, <nom-fitxer>.arff.
- **seed**: permet canviar la llavor en la qual s'agafen els valors per a fer els dos datasets train i test. Si aquest camp no s'especifica, s'agafa com a valor per defecte els últims cinc dígit del DNI d'un dels autors de la pràctica.
- **percentage**: permet canviar el percentatge de les dades que aniran al train i per tant, les del test també. Aquest percentatge s'indica de la següent manera, on si es vol un 85% l'argument a passar ha de ser 0.85. Si aquest camp no s'especifica, per defecte el percentatge serà 75%.

Dins d'aquest codi s'utilitzen dos funcions:

- **ArgumentParser**: constructor de la classe.
- **add_argument**: ens permet afegir un argument. Aquest mètode accepta diferents tipus de paràmetres per a canviar el seu comportament, els quins hem utilitzat són els següents:
 - **help**: proporciona un text d'ajuda quan es realitza la comanda **--help**.
 - **nargs**: posant **nargs** com a ? (**nargs='?'**), ens permet que si indiquem l'argument en la línia de comandes ens agafarà aquest com a únic, i a més, si aquest no està indicat s'agafarà el valor **default** com a argument.
 - **default**: el valor o l'argument que s'agafaria si aquest no ha estat indicat per la línia de comandes.
 - **type**: ens permet especificar com s'han de parsejar els valors per a poder ser utilitzats en el codi.

1.2 Pandas

Per poder realitzar el parseig de les dades hem utilitzat `pandas`. En primer lloc la funció `read_csv`, la qual ens permet llegir el dataset.

Per tractar les dades que tenim al dataset, hem canviat els noms de la columna `room_type` utilitzant el mètode `replace` per tenir-los tots amb el mateix format, ja que hi ha diferents tipus d'habitacions que el seu format és separat en guions i d'altres que es separat en espais.

També hem mapejat els valors de la columna `overall_satisfaction` on inicialment els valors d'aquesta columna eren [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5] i al mapejar-los queda de la següent manera: [1, 2, 3, 4, 5, 6, 7, 8, 9].

En les columnes `accomodates` i `bedrooms` hem canviat el tipus dels valors que hi han en aquestes columnes a string ja que volem representar aquests valors en el format `.arff`.

Les columnes `review`, `price`, `latitude` i `longitude`, per passar els seus valors continus a discrets s'ha utilitzat la funció `cut` de la llibreria `pandas`, la qual ens permet dividir el dataset en diferents intervals utilitzant el rang de valors. El n^o de divisions que s'ha escollit per aquestes columnes és de 15 ja que s'ha provat diferents n^o de divisions com per exemple, 5 o 20 i al posar els arxius train i test a Weka ens donava una *accuracy* menor comparat amb 15 divisions que ens dona major. Després d'aplicar el mètode `cut` canviem el tipus dels valors de les columnes a string ja que volem representar aquests valors en el format `.arff`.