**Array**



**Methods**

**importCars():**

Returns a list of all cars that will be used

**createAttrMatrix(carsList):**

Returns a 2d array where each column is the attribute list of a car, so rows=33 and columns=carsList.len

**getBrowsingAttrs(n):**

Returns list of n tuples (attr #, score) that comes from browsed cars and their respective times

**getPrefArr(attrs2boost):**

Returns an array of the weighted preferences associated with each attribute (prefArr.len=33)

**getScoreArray(attrMatrix, prefArr):**

Calculates score array by:

\*multiplying attributes (rows) by corresponding weighted pref values

\*summing scores for each car (column)

**getBestCars(scoreArray, n):**

Returns a list of n tuples of best cars' (indices, score)

**Data outside python files**

**CarData.xlsx**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Price | Highway MPG | Size | Color | Engine Fuel Type | Horsepower | Seats | Wheel Drive | Make |

**PrefArr.xlsx**

Contains preference values (integers -2≤x≤2) for whole array as seen above in a vertical fashion

**BrowsingAttrs.xlsx**

Contains data of cars browsed and times associated with each car

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Price | Highway MPG | Size | Color | Engine Fuel Type | Horsepower | Seats | Wheel Drive | Make | Time |

**Output**

“CarName”-“PercentMatch”% (ex: 2019 Mercedes A class sedan-87.5%)