

IDEA: VEDI VARIANTE 4 DI WORDCOUNT (metodi)

1. leggo dati e salvo in RawData, calcolo AVG considerando tutti i dati [Rating] divido per UserID
2. partiziono in k (valore predefinito) RDD
3. Map: normalizzo dati -> creo (ProductID, NormRating) tenendo tutte le entry chiave-valore, mappo solo il Rating

mapValues(f) → It transforms each key-value pair (k,v) in X into a key-value pair (k,v'=f(v)) of type Tuple2<K,V'> (with arbitrary V') where f is the function passed as a parameter. The result is a JavaPairRDD<K,V'>.

4. Reduce: raggruppamento per ProductID e calcolo MNR (max avg) per ognuno

groupByKey() → For each key k occurring in X, it creates a key-value pair (k,w) where w is an Iterable<V> containing all values of the key-value pairs with key k in X. The result is a JavaPairRDD<K,Iterable<V>>. The reduce phase of MapReduce can be implemented by applying flatMapToPair after groupByKey.

reduceByKey(f). For each key k occurring in X, it creates a key-value pair (k,v) where v is obtained by applying the commutative and associative function f passed as a parameter (e.g., (x,y)->x+y) to all values of the key-value pairs with key k in X. The result is a JavaPairRDD<K,V>.

5. Ordino per valore MNR decrescente e metto in output i primi T prodotti

sortByKey(ascending) → boolean parameter ascending, it sorts the elements of X by key

NB: oppure AVG è la media per UserID considerando ogni RDD