2η Εργασία Big Data

**Φωτιάδου Βασιλική (dai19009)**

**Μανώλη Χριστίνα (dai19181)**

# **Αναφορά**:

## Εύρεση κατάλληλου αριθμού συστάδων :

Υπολογίσαμε τον αριθμό των συστάδων (κ=6) και τα αντίστοιχα

κέντρα τους, χρησιμοποιώντας την υλοποίηση του αλγορίθμου Κ-μέσων της βιβλιοθήκης MLLib

Για εκάστοτε αρχικοποίηση κάναμε το παρακάτω πείραμα:

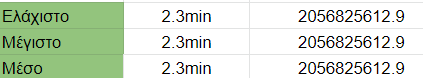
* Αρχικα δοκιμάσαμε το K για τιμές απο 5 μεχρι 20 με βήμα 5 (δηλαδή για Κ=5, Κ=10, Κ=15) με την χρήση for loop
  + Δημιουργήσαμε το kmeans μοντελο με παραμέτρους το εκάστοτε Κ και την εκάστοτε μέθοδο αρχικοποίησης
  + Κάναμε fit τα assembled δεδομένα στο μοντελο που δημιουργήσαμε πιο πάνω.
  + Αξιολογήσαμε το αποτέλεσμα του κάθε clustering χρησιμοποιώντας:
    - Το συνολικό άθροισμα των τετραγώνων των αποστάσεων.
      * cost = summary.trainingCost
    - Τον συντελεστή περιγράμματος.
      * score=evaluator.evaluate(output)
    - Το μέγεθος των συστάδων.
      * sizeCluster = summary.clusterSizes
  + Μπορείτε να δείτε τα αποτελέσματα από αυτά τα πειράματα στα randomlog1.log και parallellog1.log για έκαστο τρόπο αρχικοποιήσης.
  + Παρατηρώντας τον συντελεστή περιγράμματος και για τις δύο αρχικοποιήσεις να μειώνεται αρκετά από το Κ=10 και μετά ενώ για Κ=5 ήταν η υψηλότερη τιμή συμπεράναμε ότι το σωστό Κ θα βρίσκεται μεταξύ των τιμών 2-9.
* Στην συνέχεια, δοκιμάσαμε το K για τιμές απο 2 μεχρι 9 με βήμα 1:
  + Χρησιμοποιώντας πάλι τον ίδιο κώδικα με παραπάνω αξιολογήσαμε τα αποτελέσματα με βάση πάλι της 3 παραπάνω μετρικές και βγάλαμε τα εξής συμπεράσματα (βλ.randomlog2.log και parallellog2.log) :
    - Ο συντελεστής περιγράμματος είχε την υψηλότερη τιμή (≈0.98) για Κ = 6 ενώ ταυτόχρονα το κόστος τετραγωνικού είχε πάρει την χαμηλότερη τιμή για το ίδιο Κ. Επομένως , θεωρήσαμε το ιδανικό Κ = 6.

## Υπολογισμός χρόνου, τετραγωνικού σφάλματος με κατάλληλο Κ :

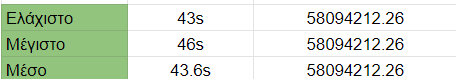
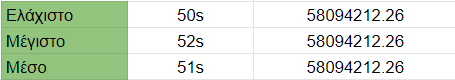
* Αφού προσδιορίσαμε το σωστό αριθμό συστάδων τρέξαμε τον αλγόριθμο KMeans για κάθε μία αρχικοποίηση και για 2 και 1 slaves υπολογίζοντας κάθε φορά το τετραγωνικό σφάλμα και καταγράφοντας τον χρόνο εκτέλεσης. Τα αποτελέσματα των χρόνων και του τετραγωνικού σφάλματος είναι τα εξής :

Χρησιμοποιώντας την εντολή $SPARK\_HOME/sbin/stop-slave.sh

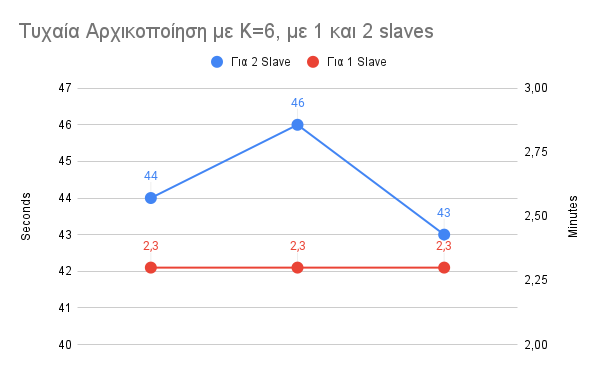
### **Για ένα slave:**

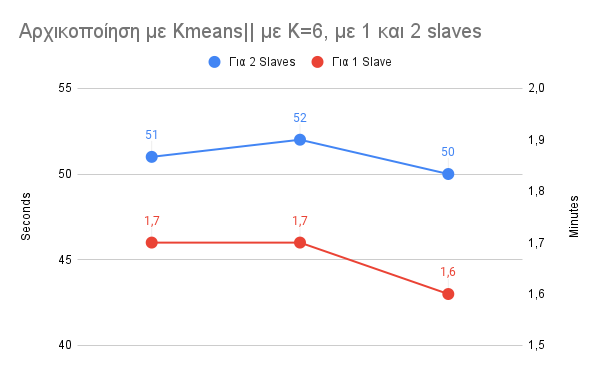
* Με αρχικοποίηση Random
* Με αρχικοποίηση KMeans||

### **Για δύο slaves:**

* Με αρχικοποίηση Random
* Με αρχικοποίηση KMeans||

## Σχολιασμός Ευρημάτων:





Συγκρίνοντας τους χρόνους για 1 slave και 2 slaves από την κάθε υλοποίηση (Τυχαία Αρχικοποίηση και Αρχικοποίηση με Kmeans||) και έχοντας θέσει το Κ=6, παρατηρούμε ότι με 2 slaves είναι αρκετά πιο γρήγορο από ότι με 1 slave και στις 2 υλοποιήσεις. Πιο συγκεκριμένα, στο 1ο διάγραμμα για Τυχαία Αρχικοποίηση θα παρατηρήσουμε ότι ο χρόνος που χρειάζεται με ένα slave είναι σχεδόν τριπλάσιος του χρόνου που χρειάζεται με 2 slaves. Αντίστοιχα, και στο 2ο διάγραμμα μπορούμε να παρατηρήσουμε ότι ο χρόνος για 1 slave είναι διπλάσιος του χρόνου που χρειάζεται με 2 slaves.