ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάθημα: Big Data/Ανάλυση Δεδομένων Μεγάλου Όγκου Εξάμηνο: Z'

Φοιτητές: Θεμιστοκλής Κουκουτζέλας-dai19022,Θεόφιλος Κιαπίδης-dai19079

2η Εργασία Μαθήματος

Εισαγωγή

Αντικείμενο της παρούσας εργασίας αποτέλεσε η ομαδοποίηση ασθενών που περιέχονται στο αρχείο meddata2022.csv που μας παρασχέθηκε, βάσει 5 τιμών από τις εξετάσεις τους. Στόχος μας ήταν η εύρεση του βέλτιστου αριθμού των συστάδων στις οποίες ομαδοποιούνται, καθώς και των μεγεθών κάθε μίας από αυτές. Για τον σκοπό αυτό χρησιμοποιήθηκε το Apache Spark και ο αλγόριθμος K-Means που περιέχεται στη βιβλιοθήκη MLlib. Ως γλώσσα προγραμματισμού χρησιμοποιήθηκε η Python.

Περιγραφή Επίλυσης

Αρχικά στόχος μας είναι να βρούμε τον βέλτιστο αριθμό Clusters, με τις τιμές των κέντρων τους. Η εργασία αυτή γίνεται στο αρχείο **assignment2.py** που παρατίθεται.

Το πρόγραμμα μας ξεκινάει με την χρήση των λειτουργιών SparkContext και SparkSession, ώστε να γίνει η απαραίτητή σύνδεση στο Spark cluster μεταξύ Master-Slave.

Έπειτα, εισάγουμε το αρχείο μας από το HDFS ,ταυτόχρονα αφαιρώντας πιθανές τιμές null,ώστε να 'ελαφρύνουμε' την επερχόμενη διαδικασία. Στη συνέχεια όλες οι τιμές θα μαζευτούν σε ένα feature vector, καθώς θα είναι ευκολότερο μέσω αυτής της δομής να εκπαιδευτεί το μοντέλο μας. Κατά αυτό τον τρόπο όμως θα προκύψουν μεγάλες διαφορές ανάμεσα στις τιμές μέσα στο vector. Για αυτό τον λόγο, θα υποστεί κανονικοποίηση, ώστε οι τιμές να κινούνται σε παρόμοια κλίμακα.

Προτού ξεκινήσουμε να δοκιμάζουμε διάφορους αριθμούς cluster, αρχικοποιούμε τον αποτιμητή που θα χρειαστεί ώστε να υπολογίζουμε κάθε φορά τον συντελεστή περιγράμματος.

Σε αυτό το σημείο ξεκινά η πραγματική δουλειά, όπου για κάθε τρόπο αρχικοποίησης

δοκιμάζουμε 10 αριθμούς clusters, από 2 έως 11, για να βρούμε την βέλτιστη διαφοροποίηση, βάσει του Sillhouette Score. Αυτό γίνεται εισάγοντας τον k-means στον κώδικα και τον εφαρμόζουμε πάνω στο feature vector, δημιουργώντας κατ' αυτό τον τρόπο

το μοντέλο μας. Υπολογίζουμε για κάθε αριθμό clusters, το Sum of Squared Errors ,το Silhouette Score, και το μέγεθος των συστάδων.

Από την παραπάνω εκτέλεση, προέκυψε πως τόσο με τυχαία αρχικοποίηση όσο και με kmeans||, ο βέλτιστος αριθμός συστάδων είναι 6 με την κάθε συστάδα να έχει μέγεθος 1.000.000 ασθενείς.

Στη συνέχεια σε ξεχωριστό πρόγραμμα**(assignment2_2.py),** στο οποίο λαμβάνονται ως arguments ο επιθυμητός αριθμός συστάδων και ο τρόπος αρχικοποίησης των κέντρων, εκτελέσαμε ακριβώς τα ίδια βήματα χωρίς for loops, ώστε να βγάλουμε διάφορα συμπεράσματα για μετρικές που προαναφέρθηκαν.

Ευρήματα

Μέθοδος Αρχικοποίησης	Αριθμός slaves	Αριθμός Εκτέλεσης	Χρόνος Εκτέλεσης	SSE
Random	2	#1	1.1 mins	50353.37757773351
Random	2	#2	1.0 mins	50353.37757773351
Random	2	#3	1.1 mins	50353.37757773351
K-means||	2	#1	1.2 mins	50353.37757773352
K-means||	2	#2	1.2 mins	50353.37757773351
K-means||	2	#3	1.2 mins	50353.37757773352
Random	1	#1	2.6 mins	1782761.3764309604
Random	1	#2	2.8 mins	1782761.3764309604
Random	1	#3	2.7 mins	1782761.3764309604
K-means||	1	#1	2.1 mins	50353.377577733045
K-means||	1	#2	2.2 mins	50353.377577733045
K-means||	1	#3	2.2 mins	50353.377577733045

Όσο αφορά τους χρόνους:

Ελάχιστος χρόνος για 1 slave με αρχικοποίηση Random: 2.6 mins

Μέγιστος χρόνος για 1 slave με αρχικοποίηση Random: 2.8 mins

Μέσος χρόνος για 1 slave με αρχικοποίηση Random: 2.7 mins

Ελάχιστος χρόνος για 1 slave με αρχικοποίηση K-means||: 2.1 mins

Μέγιστος χρόνος για 1 slave με αρχικοποίηση K-Means||: 2.2 mins

Μέσος χρόνος για 1 slave με αρχικοποίηση K-Means||:2.16 mins

Ελάχιστος χρόνος για 2 slaves με αρχικοποίηση K-means||: 1.2mins

Μέγιστος χρόνος για 2 slaves με αρχικοποίηση K-Means||: 1.2 mins

Μέσος χρόνος για 2 slaves με αρχικοποίηση K-Means||:1.2 mins

Ελάχιστος χρόνος για 2 slaves με αρχικοποίηση Random: 1.2mins

Μέγιστος χρόνος για 2 slaves με αρχικοποίηση Random: 1.2 mins

Μέσος χρόνος για 2 slaves με αρχικοποίηση Random:1.2 mins

Όσο αφορά το SSE:

Το SSE παρέμεινε σταθερό ανά περίπτωση σχεδόν σε όλες τις εκτελέσεις, εκτός από την εκτέλεση σε 2 slaves με αρχικοποίηση κέντρων K-Means|| όπου:

Ελάχιστο SSE: 50353.37757773351

Μέγιστο SSE: 50353.37757773352

Μέσο SSE : 50353.377577733516

Παρατηρούμε πως έχουμε πιο γρήγορα αποτελέσματα με χρήση 2 workers, με τους μέσους χρόνους εκτέλεσης να μην ξεπερνούν τα 1.2 λεπτά, ενώ στον 1 worker οι μέσοι χρόνοι είναι γύρω στα 2.5 λεπτά. Με 2 workers οριακά παραπάνω χρόνο χρειάζεται στην αρχικοποίηση κέντρων με k-means||. Το αντίστροφο συμβαίνει στον 1, καθώς εκεί η τυχαία αρχικοποίηση χρειάζεται παραπάνω χρόνο εκτέλεσης.

Άξιο παρατήρησης είναι επίσης το SSE, στην τυχαία αρχικοποίηση με 1 slave. Ενώ σε όλες τις υπόλοιπες περιπτώσεις προσεγγίζει περίπου το 50353.377, σε αυτή τη περίπτωση προσεγγίζει το 1782761.3764. Αυτό συμβαίνει διότι έχουν βρεθεί εντελώς διαφορετικά κέντρα σε σχέση με τις υπόλοιπες 9 εκτελέσεις και σαφώς χαμηλότερο silhouette score που προσεγγίζει το 0.59 ενώ σε όλες τις υπόλοιπες το 0.99.

Τελικές αναφορές

Μαζί με το παρόν έγγραφο παρατίθενται:

-2 αρχεία κώδικα στον φάκελο SourceCode, που εκτελούν τις εργασίες που περιγράφηκαν

-Οδηγίες εκτέλεσης των παραπάνω στο αρχείο readme.txt

-Αρχεία με τα κέντρα των συστάδων ανά αριθμό workers και μέθοδο αρχικοποίησης κέντρων,στον φάκελο Outputs

-Την έξοδο του Spark στο αρχείο CenterResults, ως προς την εκτέλεση του πρώτου προγράμματος για την εύρεση βέλτιστου αριθμού συστάδων, με παράθεση:α)Του Sum of Squared Errors ανα αριθμό κέντρων,β)Του Silhouette Score(Συντελεστή Περιγράμματος) και γ) του μεγέθους των συστάδων. Στο τέλος των επαναλήψεων για κάθε μέθοδο αρχικοποίησης κέντρων, εμφανίζει το βέλτιστο αριθμό συστάδων που βρήκε.