1ο Πρότζεκτ στο μάθημα "Διαχείριση Μεγάλων Δεδομένων"

– Ανάλυση δεδομένων για επιτυχημένους επιχειρηματίες με χρήση **HDFS** –

παραδοτέο από

Κουνάδη Βασιλική (Α.Μ.: 2022202000102)
Μαζαράκης Ιωάννης (Α.Μ.: 2022202000130)
Τετράδης Αναστάσιος (Α.Μ.: 2022202000206)

22 Δεκεμβρίου 2023

Επιβλέπων: Ραυτοπούλου Παρασκευή

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Σχολή Οικονομίας και Τεχνολογίας
Πανεπιστήμιο Πελοποννήσου

Περιεχόμενα

1 Οδηγίες εγκατάστασης

Παρακάτω υπάρχοθν αναλυτικές οδηγίες για την εκτέλεση προγραμμάτων MapReduce σε Hadoop με έτοιμα αρχεία JAR, εφόσον το Hadoop είναι προεγκατεστημένο και λειτουργικό στον υπολογιστή σας.

Βήμα 1: Λήψη του Πρότζεκτ

Κατεβάστε το έργο στον τοπικό σας υπολογιστή.

Βήμα 2: Εκκίνηση του Hadoop

Ανοίξτε το τερματικό σας και ξεκινήστε το προεγκατεστημένο Hadoop χρησιμοποιώντας τις κατάλληλες εντολές για το περιβάλλον σας.

Βήμα 3: Δημιουργία Καταλόγου Εισόδου

Δημιουργήστε ένα directory εισόδου/input στο Hadoop με την εντολή:

```
hadoop fs -mkdir /input
```

Βήμα 4: Μεταφόρτωση Αρχείου CSV

Βάλτε το αρχείο CSV που παρέχεται στο πρότζεκτ στο directory εισόδου/input που δημιουργήσατε με την εντολή:

```
hadoop fs -put yourpath/Fixed_BillionairesStatistics.csv /input
```

Βήμα 5: Επιβεβαίωση Μεταφόρτωσης CSV

Για να ελέγξετε αν το αρχείο CSV φορτώθηκε επιτυχώς, χρησιμοποιήστε την εντολή:

```
hadoop fs -ls /input/
```

Βήμα 6: Εκτέλεση Αρχείου JAR MapReduce

Εκτελέστε τα προγράμματα MapReduce εκτελώντας το αρχείο JAR με την εντολή:

```
hadoop jar yourpath/jarName.jar /input /output
```

Φροντίστε κάθε φορά να ονομάζεται αλλιώς το directory output.

Βήμα 7: Προβολή Αποτελεσμάτων

Για να δείτε τα αποτελέσματα στο directory εξόδου (που δημιούργησε πριν αυτόματα το Hadoop), χρησιμοποιήστε την εντολή:

```
hadoop fs -cat /output/*
```

Εναλλακτικά, μπορείτε να επισκεφθείτε τα αποτελέσματα στο διαδικτυο, πηγαίνοντας στο http://localhost:9870/.

Σημείωση: Βεβαιωθείτε ότι αντικαθιστάτε το "yourpath" με το πραγματικό path των αρχείων και "jarName.jar" με το όνομα του αρχείου JAR που θέλετε να εκτελέσετε.

- **q1** – Υπολογισμός του Range στο numeric attribute "age".
- **q1b** – Υπολογισμός του Range στο nominal attribute "category".
- **q2** – Σε ποια ηλικία θα καταφέρω να γίνω πλούσιος;
- **q3** – Σε ποια χώρα να ζω τέλος πάντων για να γίνω πλούσιος;
- **q4** – Βοηθητικός υπολογισμός totalWorth για χρήση στο q4b.
- **q4b** – Με ποιον τομέα πρέπει να ασχοληθώ για να πιάσω την καλή;

2 Επεξεργασία των δεδομένων

Η προεπεξεργασία των δεδομένων περιλάμβανε κυρίως τη διαγραφή της πρώτης γραμμής από το αρχείο CSV, δουλεύοντας στο διορθωμένο αρχείο "Fixed_BillionersStatistics.csv" μέσα στον φάκελο του πρότζεκτ. Η υπόλοιπη επεξεργασία των δεδομένων έγινε μέσα στα προγράμματα Java, ειδικότερα μέσα στις μεθόδους Map που αναλαμβάνουν τον χειρισμό του αρχείου CSV.

- Χειρισμός Κενών Τιμών Χαρακτηριστικών
  Οι κενές τιμές σε χαρακτηριστικά (π.χ., "κάτι, κάτι, ,κάτι, ...") ο χειρισμός τους περιλάμβανε μια if statement. Όταν χρησιμοποιείται η συνάρτηση split() στη Java με το κόμμα ως διαχωριστικό, ένα κενό string ("") μπαίνει στα κελιά του πίνακα string (που προκύπτει μετά τη split()) όταν δεν υπάρχει τίποτα μεταξύ των κομμάτων. Έτσι, η συνάρτηση Map παραλείπει την εκπομπή εάν η τιμή που έχει είναι ένα κενό string.

- Χειρισμός Περισσότερων Κομμάτων Εντός Εισαγωγικών
  Ορισμένα χαρακτηριστικά περιείχαν έξτρα κόμματα μέσα σε εισαγωγικά (πχ. για ονόματα οικογενειών). Για να αντιμετωπίσουμε αυτό το πρόβλημα κατά τη διαδικασία του split(), χρησιμοποιήσαμε ένα regex αντί για ένα απλό κόμμα. Αυτό το regex αγνοεί τα κόμματα που βρίσκονται μέσα σε εισαγωγικά, εξασφαλίζοντας ότι ο διαχωρισμός γίνεται μόνο στα κανονικά κόμματα έξω από τα εισαγωγικά, χωρίζοντας σωστά τα χαρακτηριστικά.

- Χειρισμός του Χαρακτηριστικού "cpi_country"
  Το χαρακτηριστικό "cpi_country" (το 25ο από τα 35) που απαιτείται στο 5, αποδείχθηκε προβληματικό, καθώς δεν έχει πάντα τιμή. Αφού μελετήσαμε τα δεδομένα, διαπιστώσαμε ότι υπάρχουν εγγραφές με 20, 25 ή 35 χαρακτηριστικά. Οι εγγραφές με 20 χαρακτηριστικά δεν περιλαμβάνουν το "cpi", οπότε τις παραλείπουμε πλήρως. Χρησιμοποιούμε μόνο τις εγγραφές με 25 ή 35 χαρακτηριστικά, εξασφαλίζοντας την παρουσία του "cpi_country".

3 Μελέτη των χαρακτηριστικών

3.1 Τύπος των χαρακτηριστικών/attributes

Προτού ξεκινήσουμε να αναλύουμε τα δεδομένα μας, είναι σημαντικό να τα μελετήσουμε και να κατανοήσουμε τους τύπους πληροφορίας που περιέχει το σύνολο δεδομένων μας. Ταξινομούμε τα χαρακτηριστικά σε δύο κύριες κατηγορίες: τα numeric και τα nominal. Τα numeric χαρακτηριστικά είναι είτε interval είτε ratio-scaled, υποδεικνύοντας τον τρόπο μέτρησής τους. Τα nominal χαρακτηριστικά περιλαμβάνουν binary και ordinal τύπους, .δίνοντάς μας πληροφορίες για τις κατηγορίες.

- Το χαρακτηριστικό "rank" είναι τύπου ordinal (nominal), καθώς, παρόλο που οι τιμές που παίρνει είναι αριθμοί, αντιπροσωπεύουν την κατάταξη των εκατομμυριούχων ως προς τον πλούτο τους σε φθίνουσα σειρά (έχει την χρήση ενός ranking συστήματος, και όχι ενός απλού αριθμού id).

- Το χαρακτηριστικό "finalWorth" είναι τύπου ratio-scaled (numeric) , καθώς οι τιμές του αντιπροσωπεύουν την χρηματική αξία ενός ατόμου σε δολάρια. Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "category" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας κατηγορία ή βιομηχανία στην οποία λειτουργεί η επιχείρηση του δισεκατομμυριούχου, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "personName" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας τα ονόματα των εκατομμυριούχων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "age" είναι τύπου ratio-scaled (numeric) , καθώς οι τιμές του αντιπροσωπεύουν την ηλικία ενός ατόμου. Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "country" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα χωρών, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "city" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα πόλεων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "source" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας τα ονόματα των ανθρώπων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "industries" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα ή τίτλους (πχ. Εταιρίες, βιομηχανίες) για την πηγή εισοδημάτων του εκάστοτε εκατομμυριούχου, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "countryOfCitizenship" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα χωρών, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "organization" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα ή τίτλους εταιριών που σχετίζονται με τον εκάστοτε εκατομμυριούχο, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "selfMade" είναι τύπου binary (nominal), καθώς αναφέρεται μόνο σε δύο διακριτικές ονομαστικές κατηγορίες, χωρίς κάποια συγκεκριμένη σειρά ή άλλη διάκριση μεταξύ τους. Συγκεκριμένα, οι δύο μόνο δυνατές τιμές που μπορεί να πάρει είναι "true" και "false", αντιπροσωπεύοντας το αν ο εκατομμυριούχος είναι κληρονόμος του πλούτου του ή όχι.

- Το χαρακτηριστικό "status" είναι τύπου binary (nominal), καθώς αναφέρεται μόνο σε δύο διακριτικές ονομαστικές κατηγορίες, χωρίς κάποια συγκεκριμένη σειρά ή άλλη διάκριση μεταξύ τους. Συγκεκριμένα, οι δύο μόνο δυνατές τιμές που μπορεί να πάρει είναι "D" και "U", αντιπροσωπεύοντας το αν ο εκατομμυριούχος είναι κληρονόμος του πλούτου του ή όχι.

- Το χαρακτηριστικό "gender" είναι τύπου n binary (nominal), καθώς αναφέρεται μόνο σε δύο διακριτικές ονομαστικές κατηγορίες, χωρίς κάποια συγκεκριμένη σειρά ή άλλη διάκριση μεταξύ τους. Συγκεκριμένα, οι δύο μόνο δυνατές τιμές που μπορεί να πάρει είναι "male" και "female", αντιπροσωπεύοντας το φύλλο του εκατομμυριούχου.

- Το χαρακτηριστικό "birthDate" είναι τύπου ordinal (nominal), καθώς οι τιμές του δεν είναι αριθμοί (DD/MM/YYYY), άρα εν μέρη ανήκουν σε διάφορες ≪ονομαστικές≫ κατηγορίες, και αντιπροσωπεύουν την ημερομηνία γέννησης ενός ατόμου. Αυτό σημαίνει ότι υπάρχει μια σειρά ή μια διάκριση μεταξύ των ημερομηνιών (πχ. ποιος γεννήθηκε πρώτος), καθώς μπορούμε να πούμε ποια ημερομηνία προηγείται ή ακολουθεί μια άλλη.

- Το χαρακτηριστικό "lastName" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας τα επώνυμα των εκατομμυριούχων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "firstName" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας τα μικρά ονόματα των εκατομμυριούχων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "title" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας τους πιθανούς τιμητικούς τίτλους των εκατομμυριούχων, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "date" είναι τύπου ordinal (nominal), καθώς οι τιμές του δεν είναι αριθμοί (DD/MM/YYYY), άρα εν μέρη ανήκουν σε διάφορες «ονομαστικές» κατηγορίες, και αντιπροσωπεύουν την ημερομηνία καταγραφής των δεδομένων. Αυτό σημαίνει ότι υπάρχει μια σειρά ή μια διάκριση μεταξύ των ημερομηνιών (πχ. ποια καταγραφή έγινε ποιο πρόσφατα), καθώς μπορούμε να πούμε ποια ημερομηνία προηγείται ή ακολουθεί μια άλλη.

- Το χαρακτηριστικό "state" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα πολιτειών, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους. Το χαρακτηριστικό "residenceStateRegion" είναι τύπου nominal, καθώς οι τιμές που μπορεί να πάρει ανήκουν σε διάφορες ονομαστικές κατηγορίες, αντιπροσωπεύοντας ονόματα περιοχών, χωρίς κάποια συγκεκριμένη σειρά ή διάκριση μεταξύ τους.

- Το χαρακτηριστικό "birthYear" είναι τύπου interval (numeric) , καθώς οι τιμές του αντιπροσωπεύουν έτη γεννήσεως. Άρα, οι τιμές του έχουν σειρά και μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους. Ωστόσο, δεν υπάρχει πραγματικό μηδενικό, καθώς το μηδέν δεν υποδηλώνει απόλυτη έλλειψη ενός έτους, και μπορούμε να συγκρίνουμε τα έτη μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "birthMonth" είναι τύπου interval (numeric), καθώς οι τιμές του αντιπροσωπεύουν μήνες γεννήσεως. Άρα, οι τιμές του έχουν σειρά και μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους. Ωστόσο, δεν υπάρχει πραγματικό μηδενικό, καθώς το μηδέν δεν υποδηλώνει απόλυτη έλλειψη ενός μήνα, και μπορούμε να συγκρίνουμε τους μήνες μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "birthDay" είναι τύπου interval (numeric), καθώς οι τιμές του αντιπροσωπεύουν ημέρες γεννήσεως. Άρα, οι τιμές του έχουν σειρά και μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους. Ωστόσο, δεν υπάρχει πραγματικό μηδενικό, καθώς το μηδέν δεν υποδηλώνει απόλυτη έλλειψη μιας ημέρας, και μπορούμε να συγκρίνουμε τις ημέρες μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "cpi_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν τον Δείκτης Τιμών Καταναλωτή μιας χώρας (δηλαδή

χρηματικό ποσό). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "cpi_change_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν την αλλαγή του Δείκτη Τιμών Καταναλωτή μιας χώρας (δηλαδή ποσοστό). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "gpd_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν τον Ακαθάριστο Εγχώριο Προϊόν μιας χώρας (δηλαδή χρηματικό ποσό). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "gross_tertiary_education_enrollment" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν τον αριθμό ατόμων στην τριτοβάθμια εκπαίδευση σε μια χώρα. Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "gross_primary_education_enrollment_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν τον αριθμό ατόμων στην πρωτοβάθμια εκπαίδευση σε μια χώρα. Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "life_expectancy_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν το προσδόκιμο ζωής μιας χωράς (μια ηλικία). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "tax_revenue_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν την ατομική φορολογία μιας χώρας (ένα ποσοστό). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "total_tax_rate_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν την συνολική φορολογία μιας χώρας (ένα ποσοστό). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "population_country" είναι τύπου ratio-scaled (numeric), καθώς οι τιμές του αντιπροσωπεύουν τον πληθυσμό μιας χωράς (αριθμός ατόμων). Άρα, οι τιμές του έχουν σειρά, μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους, έχουν μηδενικό σημείο (το οποίο υποδηλώνει μηδενική ποσότητα ή πλήρη έλλειψη του αντικειμένου που μετράμε) και μπορούμε τις συγκινούμε μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "latitude_country" είναι τύπου interval (numeric), καθώς οι τιμές του αντιπροσωπεύουν γεωγραφικά πλάτη. Άρα, οι τιμές του έχουν σειρά και μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους. Ωστόσο, δεν υπάρχει πραγματικό μηδενικό, καθώς το μηδέν δεν υποδηλώνει απόλυτη έλλειψη γεωγραφικού πλάτους, και μπορούμε να συγκρίνουμε τα πλάτη μεταξύ τους ποσοτικά.

- Το χαρακτηριστικό "longitude_country" είναι τύπου interval (numeric), καθώς οι τιμές του αντιπροσωπεύουν γεωγραφικά μήκη. Άρα, οι τιμές του έχουν σειρά και μετρούνται σε κλίμακα με μονάδες ίσου μεγέθους. Ωστόσο, δεν υπάρχει πραγματικό μηδενικό, καθώς το μηδέν δεν υποδηλώνει απόλυτη έλλειψη γεωγραφικού μήκους, και μπορούμε να συγκρίνουμε τα μήκη μεταξύ τους ποσοτικά.

3.2 Διαφορές στην αντιμετώπιση των χαρακτηριστικών

Η ανάλυση τόσο των numeric όσο και των nominal χαρακτηριστικών είναι απαραίτητη για την κατανόηση των δεδομένων μας. Για το ερώτημα, έχουμε επιλέξει το numeric χαρακτηριστικό "age" και το nominal χαρακτηριστικό "category". Ο στόχος είναι να βρούμε το εύρος τιμών για κάθε χαρακτηριστικό, λαμβάνοντας υπόψιν τις τις εγγενείς διαφορές στην αντιμετώπιση αυτών των τύπων χαρακτηριστικών.

Απ' τη μια, για το numeric χαρακτηριστικό "age", ο υπολογισμός του εύρους (range) είναι μια απλή διαδικασία. Αφαιρώντας την ελάχιστη ηλικία από τη μέγιστη ηλικία (max-min) που βρήκαμε στο σύνολο δεδομένων, λαμβάνουμε μια αριθμητική αναπαράσταση της εξάπλωσης (spread) ή της διασποράς (dispersion) των τιμών ηλικίας. Η οπτικοποίηση αυτών των πληροφοριών μέσω ενός boxplot παρέχει μια ολοκληρωμένη εικόνα της κατανομής, αποκαλύπτοντας όχι μόνο την κεντρική τάση (central tendency) αλλά και τις ακραίες τιμές (outliers) στα δεδομένα.
Απ' την άλλη, τα nominal χαρακτηριστικά, όπως η "category", παρουσιάζουν επιπλέον δυσκολία. Αυτές οι τιμές είναι ονόματα/λέξεις και δεν μπορούν να δεχθούν αριθμητικές πράξεις, προκειμένου να γίνει αφαίρεση (max-min) , και ακόμα, δεν μπορούμε να καθορίσουμε max και min τιμές σε ονομαστικές κατηγορίες. Άρα, ο παραδοσιακός υπολογισμός εύρους (max-min) δεν ισχύει. Έτσι, χρησιμοποιούμε μια μέθοδο που βασίζεται στη συχνότητα για να κατανοήσουμε την κατανομή των κατηγοριών στο σύνολο δεδομένων. Μετρώντας τις εμφανίσεις κάθε ξεχωριστής κατηγορίας και οπτικοποιώντας αυτές τις πληροφορίες μέσω ενός barchart, αποκτούμε πληροφορίες για την «επικράτηση» διαφορετικών κατηγοριών στα δεδομένα μας. Αυτή η μέθοδος μας επιτρέπει να επισημάνουμε την ποικιλομορφία και την κατανομή του ονομαστικού χαρακτηριστικού, παρουσιάζοντας τη συχνότητα κάθε κατηγορίας και όχι ένα αριθμητικό εύρος.
Συνοπτικά, ενώ στα numeric χαρακτηριστικά μπορούμε να κάνουμε για ποσοτικές μετρήσεις, όπως υπολογισμούς εύρους και boxplot, τα ονομαστικά χαρακτηριστικά απαιτούν εναλλακτικές μεθόδους που επικεντρώνονται στις συχνότητες και τις κατανομές των κατηγοριών.

3.3 Υπολογισμός του Range στο nominal attribute "category"

3.3.1 Ψευδοκώδικας

Ο σκοπός της μεθόδου Map είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, εξάγοντας το χαρακτηριστικό "category" και να εκπέμπει ζεύγη key-value. Κάθε ζεύγος αποτελείται από το key, ως η τιμή του "category" στη συγκεκριμένη εγγραφή, διότι στη συνέχεια θέλουμε να γκρουπ-άρουμε όλες τις εγγραφές με κοινό key (δηλαδή κοινή κατηγορία), προκειμένου να τις επεξεργαστούμε μαζί. Ως value, εκπέμπεται η αντίστοιχη τιμή 1, προκειμένου να καταμετρηθεί η συχνότητα εμφάνισης της εκάστοτε κατηγορίας.

Αλγόριθμος 1: Ψευδοκώδικας για Mapping()

```
class Mapper
    method Map (docID d, null)
        foreach line l in d
            emit (category c, value 1)
```

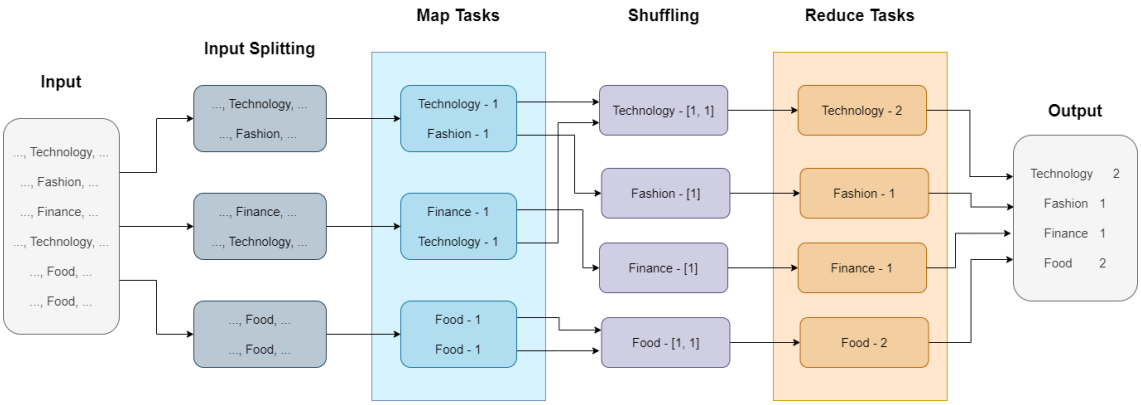
Η μέθοδος Reduce χειρίζεται το key "category" και την σχετική λίστα άσων/εμφανίσεων που έχει προκύψει. Υπολογίζει το συνολικό άθροισμα των άσων που αντιστοιχούν στη συγκεκριμένη κατηγορία. Έπειτα, εκπέμπει το key "category" και το value το οποίο αντιστοιχεί στο συνολικό άθροισμα των άσων/εμφανίσεων. Η διαδικασία αυτή επιτρέπει την εύρεση του συνολικού αριθμού εμφάνισης κάθε κατηγορίας στα δεδομένα μας, δηλαδή το range εύρος του nominal χαρακτηριστικού.

Αλγόριθμος 2: Ψευδοκώδικας για Reducing()

```
class Reducer
    method Reduce (category c, list times[])
        sum = 0
        foreach t in times[]
            sum = sum + t
        emit (category c, sum)
```

3.3.2 Σχηματική εκτέλεση

Για να κάνουμε τα MapReduce tasks και τα αποτελέσματα τους πιο εύκολα στην κατανόηση, υλοποιούμε ένα απλό και σύντομο παράδειγμα, σε μορφή διαγράμματος, πάνω σε περιορισμένες εγγραφές. Ώστε να καλύψουμε τις περισσότερες τις περιπτώσεις και να μια ρεαλιστική εικόνα/μικρογραφία του συστήματος μας, έχουμε συμπεριλάβει την ίδια τιμή σε 2 εγγραφές που βρίσκονται και εντός του ίδιου μπλοκ και σε διαφορετικά μπλοκς δεδομένων. Το παράκατω διάγραμμα μπορείτε να το βρείτε στο Examples_Schemas/q1aschema.png.



Εικόνα 1: Παράδειγμα και Σχηματική υλοποίηση της διαδικασίας MapReduce για την εύρεση όλων των κατηγοριών και των φορών εμφάνισης τους

3.3.3 Αποτελέσματα

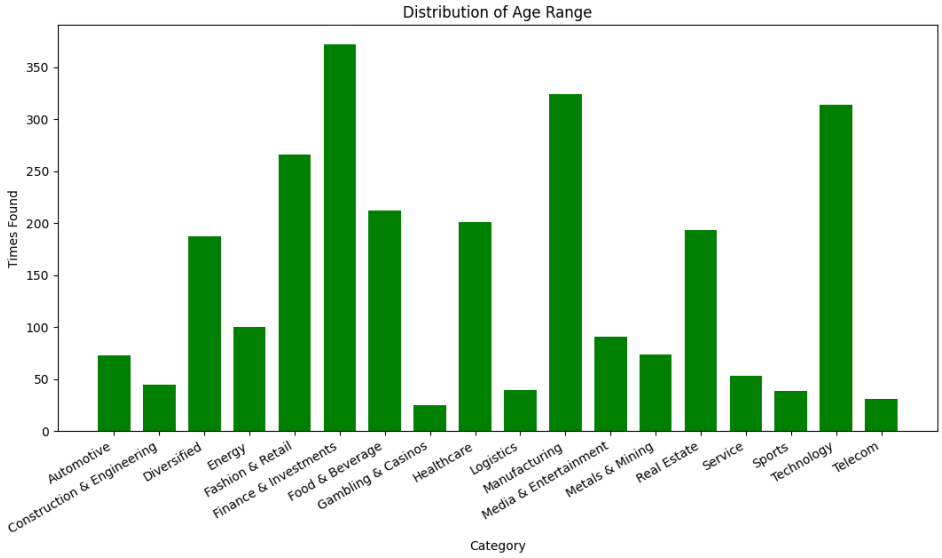
Τα αποτελέσματα που έχουμε ως file αρχείο, τρέχοντας τον αντίστοιχο κώδικα Java στο HDFS, αναπαριστόνται παράκατω σε μορφή πίνακα (για λόγους καλύτερης οργάνωσης). Το file αρχείο υπάρχει στον φάκελο του πρότζεκτ υπό το όνομα "Result_files/Q1a".

Πίνακας 1: Κατανομή κατηγοριών

Κατηγορία	**Συχνότητα**
Automotive	73
Construction & Engineering	45
Diversified	187
Energy	100
Fashion & Retail	266
Finance & Investments	372
Food & Beverage	212
Gambling & Casinos	25
Healthcare	201
Logistics	40
Manufacturing	324
Media & Entertainment	91
Metals & Mining	74
Real Estate	193
Service	53
Sports	39
Technology	314
Telecom	31

3.3.4 Γραφική απεικόνιση και Συμπεράσματα

Προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα bar graph, μεταξύ των κατηγοριών και εκατομμυριούχων που δρουν σε αυτές. Η επιλογή bar chart έγινε καθώς οι κατηγορίες στον οριζόντιο άξονα αντιπροσωπεύουν ένα nominal χαρακτηριστικό, που σημαίνει ότι δεν υπάρχει κάποια εγγενής σειρά ή κλίμακα μεταξύ τους (σε αντίθεση με του αριθμούς σε ένα numeric χαρακτηριστικό που αυξάνονται όσο προχωράμε στον άξονα).



Εικόνα 2: Bar graph κατηγορίας-συχνότητας

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε συμπεράσματα ως προς τις «προτιμήσεις» των εκατομμυριούχων σε κατηγορίες/τομείς απασχόλησης. Ο χώρος των Χρηματοοικονομικών (Finance & Investments) εμφανίζεται ως ο συχνότερος τομέας απασχόλησης, κατέχοντας τον υψηλότερο αριθμό εκατομμυριούχων, αναδεικνύοντας την τάση συσσώρευσης πλούτου στη χρηματοοικονομική βιομηχανία των μετοχών. Η Τεχνολογία (Technology) και η Παραγωγή προϊόντων (Manufacturing) παίζουν επίσης σημαντικούς ρόλους, εκπροσωπώντας την παραγωγή πλούτου μέσω της καινοτομίας και μοντέρνων τάσεων. Έπειτα ακολουθούν η βιομηχανία της Μόδας (Fashion & Retail) και της Διαχείρισης ακινήτων (Real Estate) με σημαντικές συμμετοχές. Η χαμηλή παρουσία στον χώρο του Τζόγου (Gambling & Casinos) υποδηλώνει μια πιο ιδιόμορφη κατανομή του πλούτου σε αυτόν τον τομέα, με λίγα, αλλά αρκετά πλούσια άτομα να συμμετέχουν.

3.4 Υπολογισμός του Range στο numeric attribute "age"

3.4.1 Ψευδοκώδικας

Ο σκοπός της μεθόδου Map είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, εξάγοντας το χαρακτηριστικό "age" και εκπέμποντας ζεύγη key-value. Αυτά τα ζεύγη αποτελούνται από το key "age" ως απλό string (καθώς θέλουμε να γκρουπ-άρουμε όλες τις ηλικίες μαζί με κοινό key, ώστε στο επόμενο βήμα να τις συγκρίνουμε όλες μεταξύ τους) και ως value την αντίστοιχη τιμή ηλικίας του χαρακτηριστικού στην εγγραφή.

Αλγόριθμος 3: Ψευδοκώδικας για Mapping()

```
class Mapper
    method Map (docID d, null)
        foreach line I in d
            emit (key "age", age a)
```

Η μέθοδος Reduce χειρίζεται το key "age" και την αντίστοιχη λίστα ηλικιών που έχει προκύψει. Υπολογίζει τις τιμές min, το max, το range, τα quartiles (Q1 και Q3), το median, το interquartile range (IQR) και ορίζει τα άνω και κάτω όρια για τις τιμές, προκειμένου να βρει τις ακραίες τιμές outliers. Τέλος, εκπέμπει τα παραπάνω αποτελέσματα, δηλαδή το 5 number summary του χαρακτηριστικού age. Βασικός στόχος ήταν η εύρεση του εύρους τιμών range, παρόλα αυτά θεωρήσαμε χρήσιμο να υπολογίσουμε και τις υπόλοιπες τιμές, για μελλοντική χρήση τους στον σχεδιασμό ενός boxplot για το χαρακτηριστικό age.

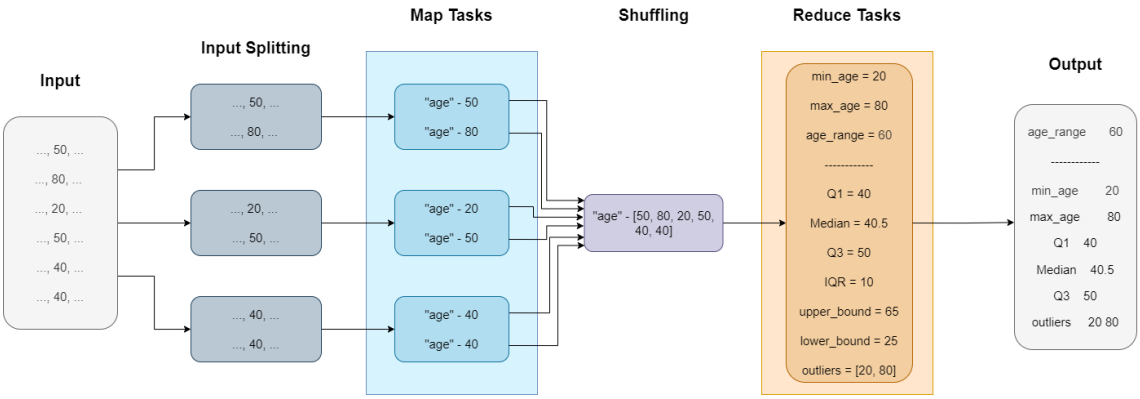
Αλγόριθμος 4: Ψευδοκώδικας για Reducing()

```
class Reducer
    method Reduce (key "age", list ages[])
        min = 0
        max = infinity
        Sort ages[] in ascending order
        foreach a in ages []
            if a<min
                min = a
```

```
        if a > max
            max=a
range = max-min
q1 = value at position (ages.length/4)
q3 = value at position (ages.length * 3/4 )
if (ages.length is odd)
    median = value at position (ages.length/2 )
else
    median = {value at position (ages.length/2 )
    + value at position ((ages.length/2) + 1)}/2
iqr= q3-q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
emit ("range", range)
emit ("min", min)
emit ("max", max)
emit ("q1", q1)
emit ("q3", q3)
emit ("median", median)
foreach a in ages[]
    if (a < lower_bound || a > upper_bound)
        emit ("outlier", a)
```

3.4.2 Σχηματική εκτέλεση

Για να κάνουμε τα MapReduce tasks και τα αποτελέσματα τους πιο εύκολα στην κατανόηση, υλοποιούμε ένα απλό και σύντομο παράδειγμα, σε μορφή διαγράμματος, πάνω σε περιορισμένες εγγραφές. Ώστε να καλύψουμε τις περισσότερες τις περιπτώσεις και να μια ρεαλιστική εικόνα/μικρογραφία του συστήματος μας, έχουμε συμπεριλάβει την ίδια τιμή σε 2 εγγραφές που βρίσκονται και εντός του ίδιου μπλοκ και σε διαφορετικά μπλοκς δεδομένων. Το παράκατω διάγραμμα μπορείτε να το βρείτε στο Examples_Schemas/q1bschema.png.



Εικόνα 3: Παράδειγμα και Σχηματική υλοποίηση της διαδικασίας MapReduce για την εύρεση του 5-number-summary

3.4.3 Αποτελέσματα

Τα αποτελέσματα που έχουμε ως file αρχείο, τρέχοντας τον αντίστοιχο κώδικα Java στο HDFS, αναπαριστόνται παράκατω σε μορφή πίνακα (για λόγους καλύτερης οργάνωσης). Το file αρχείο υπάρχει στον φάκελο του πρότζεκτ υπό το όνομα "Result_files/Q1b".

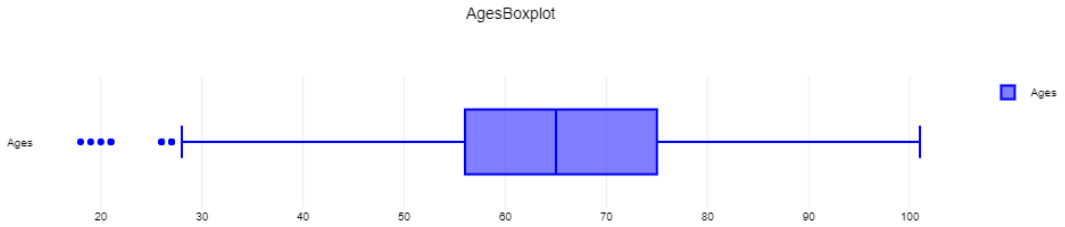
Πίνακας 2: 5-number-summary ηλικίας

Στατιστικά	**Τιμή**
Min	18
Max	101
Range	83
Q1	56
Median	65.0
Q3	75
Lower Bound	27.5
Upper Bound	103.5
Outliers	18 19 20 21 21 26 26 27 27

3.4.4 Γραφική απεικόνιση και Συμπεράσματα

Προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα boxplot, εφόσον βασικό ζητούμενο ήταν να βρούμε το range της ηλικίας. Η επιλογή boxplot έγινε καθώς είχαμε μόνο ένα χαρακτηριστικό και όχι κάτι να το συγκρίνουμε, και επιπλέον θέλαμε να δούμε σχηματικά και την κατανομή των ηλικιών εντός των εκατομμυριούχων.

Να σημειωθεί επίσης ότι εφόσον η ηλικίες 18, 19, 20, 21, 26, 27 είναι outliers, παρόλο που το 18 είναι η πραγματική minimum τιμή, για το boxplot θα ορίσουμε minimum τιμή το 28 για το κάτω whisker.



Εικόνα 4: Boxplot ηλικίας

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε συμπεράσματα ως προς την κατανομή και τα χαρακτηριστικά των εκατομμυριούχων, αν και τα παρακάτω αποτελέσματα ήταν αναμενόμενα και συμφωνούν με τις υποθέσεις μας. Το εύρος, από 18 έως 101 χρόνια, υποδεικνύει ένα ευρύ φάσμα ηλικιών, δείχνοντας την ποικιλία μεταξύ των εκατομμυριούχων. Το κάτω τεταρτημόριο (Q1) στα 56 και το πάνω τεταρτημόριο (Q3) στα 75, μαζί με τη μέση τιμή (median) στα 65, δείχνουν μια συμμετρική κατανομή (symmetric skewed), παρόλο που μοιάζει σαν θετική (positively skewed) μάλιστα –αφού η τιμή median είναι πιο κοντά στη τιμή του Q1, παρά στου Q3 ή ακριβώς στη μέση, αλλά πρόκειται για αμελητέα διαφορά μιας μονάδας-, και γενικότερα φαίνεται η συγκέντρωση των εκατομμυριούχων προς τις μεγαλύτερες ηλικιακές ομάδες. Αυτό συμφωνεί με την λογική ότι η συσσώρευση πλούτου απαιτεί χρόνο. Οι ακραίες τιμές (outliers), ιδιαίτερα στη νεότερη ηλικιακή ομάδα (18 έως 27), υποδεικνύει ότι μερικοί εκατομμυριούχοι είναι αρκετά μακριά από την τυπική κατανομή των ηλικιών, πιθανόν λόγω ξαφνικών επιτυχιών νωρίς ή κληρονομημένο πλούτο.

4 Σε ποια ηλικία θα καταφέρω να γίνω πλούσιος;

4.1 Ψευδοκώδικας

Ο σκοπός της μεθόδου Map είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, ελέγχοντας και εξάγοντας την τιμή του χαρακτηριστικού "age" και να εκπέμπει ζεύγη

key-value. Κάθε ζεύγος έχει ως key ένα string που αναπαριστά την ηλικιακή ομάδα (π.χ., "1524"), και ως value την αντίστοιχη ηλικία (που ανήκει σε αυτή την ηλιακή ομάδα/πληρεί την προϋπόθεση της ανάλογης if statement). Αυτό γίνεται προκειμένου να γκρουπ-άρουμε τις εγγραφές που ανήκουν σε κοινή ηλιακή ομάδα για μετέπειτα επεξεργασία.

Αλγόριθμος 5: Ψευδοκώδικας για Mapping()

```
class Mapper
    method Map (docID d, null)
        foreach line I in d
        if (age a > 15 && age a < 24)
            emit (key "15-24", age a)
        else if (age a > 25 && age a < 34)
            emit (key "25-34", age a)
        else if (age a > 35 && age a < 44)
            emit (key "35-44", age a)
        else if (age a > 45 && age a < 54)
            emit(key "45-54", age a)
        else if (age a > 55 && age a < 64)
            emit(key "55-64", age a)
        else if (age a > 65 && age a < 74)
            emit(key "65-74", age a)
        else if (age a > 75 && age a < 84)
            emit(key "75-84", age a)
        else if (age a > 85 && age a < 94)
            emit(key "85-94", age a)
        else if (age a > 95 && age a < 104)
            emit(key "95-104", age a)
```

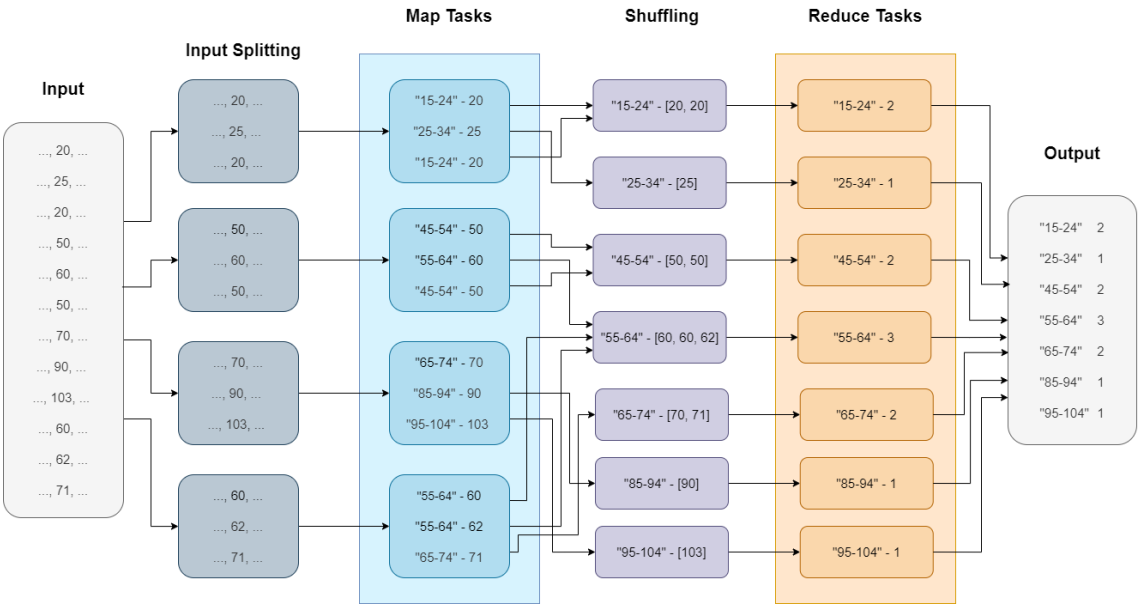
Η μέθοδος Reduce διαχειρίζεται το key "age group" και την λίστα των αντίστοιχων ηλικιών. Υπολογίζει τον συνολικό αριθμό των εγγραφών σε κάθε ηλικιακή ομάδα, βλέποντας το μήκος της λίστας value για κάθε ηλικιακή ομάδα, και εκπέμπει το key "age group" με αυτή τη συνολική τιμή. Η διαδικασία αυτή επιτρέπει τον υπολογισμό του συνολικού αριθμού ατόμων που ανήκουν σε κάθε ηλικιακή ομάδα στα δεδομένα μας.

Αλγόριθμος 6: Ψευδοκώδικας για Reducing()

```
class Reducer
    method Reduce (key age_group, list count[])
        emit(key age_group, count.length)
```

4.2 Σχηματική εκτέλεση

Για να κάνουμε τα MapReduce tasks και τα αποτελέσματα τους πιο εύκολα στην κατανόηση, υλοποιούμε ένα απλό και σύντομο παράδειγμα, σε μορφή διαγράμματος, πάνω σε περιορισμένες εγγραφές. Ώστε να καλύψουμε τις περισσότερες τις περιπτώσεις και να μια ρεαλιστική εικόνα/μικρογραφία του συστήματος μας, έχουμε συμπεριλάβει την ίδια τιμή σε 2 εγγραφές που βρίσκονται και εντός του ίδιου μπλοκ και σε διαφορετικά μπλοκς δεδομένων. Το παράκατω διάγραμμα μπορείτε να το βρείτε στο Examples_Schemas/q2schema.png.



Εικόνα 5: Παράδειγμα και Σχηματική υλοποίηση της διαδικασίας MapReduce για την εύρεση του πλήθους ατόμων ανά ηλικιακή ομάδα

4.3 Αποτελέσματα

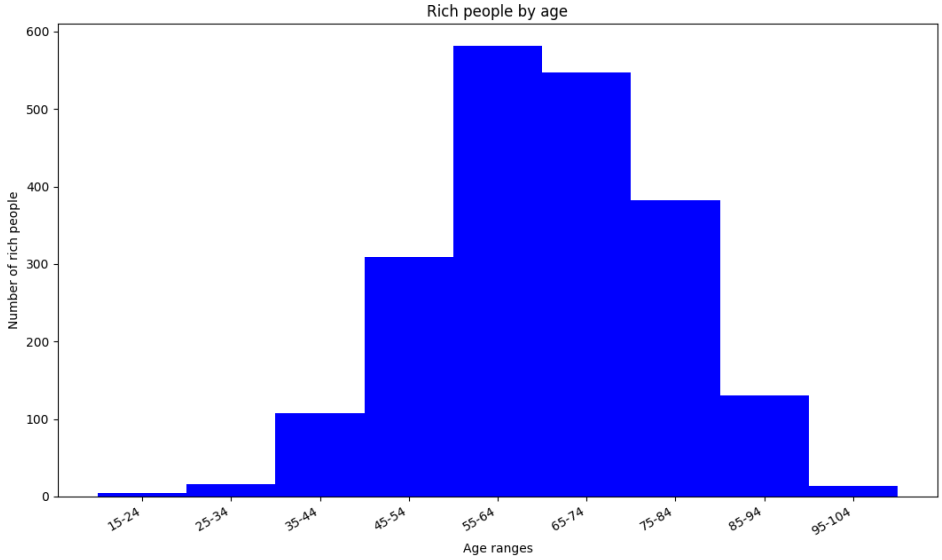
Τα αποτελέσματα που έχουμε ως file αρχείο, τρέχοντας τον αντίστοιχο κώδικα Java στο HDFS, αναπαριστόνται παράκατω σε μορφή πίνακα (για λόγους καλύτερης οργάνωσης). Το file αρχείο υπάρχει στον φάκελο του πρότζεκτ υπό το όνομα "Result_files/Q2".

Πίνακας 3: Κατανομή ηλικιακών ομάδων

Ηλικιακή ομάδα	**Συχνότητα**
15-24	5
25-34	16
35-44	107
45-54	309
55-64	581
65-74	547
75-84	382
85-94	130
95-104	14

4.4 Γραφική απεικόνιση και Συμπεράσματα

Προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα histogram, μεταξύ των ηλικιακών ομάδων και εκατομμυριούχων που δρουν σε αυτές. Η επιλογή histogram έγινε καθώς οι κατηγορίες στον οριζόντιο άξονα, ενώ μπορεί να είναι κατηγορίες με όνομα, αντιπροσωπεύουν ένα numeric χαρακτηριστικό, την ηλικία, που αυξάνονται όσο προχωράμε στον άξονα και μετράται με κοινή μονάδα.



Εικόνα 6: Histogram ηλικιακής ομάδας-συχνότητας

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε συμπεράσματα ως προς την ηλικιακή κατανομή των εκατομμυριούχων, αν και τα παρακάτω αποτελέσματα ήταν αναμενόμενα και συμφωνούν με τις υποθέσεις μας, και αντιστοιχούν με αυτά στο 3.4.4. Οι κορυφές στις ηλικιακές ομάδες, των 55-64 και 65-74 ετών, υποδεικνύουν ότι ένας σημαντικός αριθμός ατόμων φτάνει στο καθεστώς του εκατομμυριούχου κατά τη διάρκεια της μέσης ή τρίτης ηλικίας. Το διάγραμμα παρουσιάζει μια ελαφριά αρνητική κατανομή (negatively skewed), με το μεγαλύτερο πλήθος εκατομμυριούχων να συσσωρεύονται στα δεξιά του, δηλαδή στις μεγαλύτερες ηλικίες. Η ραγδαία αύξηση του αριθμού εκατομμυριούχων μαζί με την ηλικία, βεβαιώνει την ιδέα ότι η περιουσία να αυξάνεται με τον χρόνο. Αντιθέτως, υπάρχουν λίγες μετρήσεις στις πολύ ηλικιωμένες ηλικιακές ομάδες (85-94 και 95-104), δείχνοντας την λογική μείωση του αριθμού των εκατομμυριούχων σε προχωρημένη ηλικία, κυρίως λόγω θνησιμότητας. Η χαμηλότερη μέτρηση στην ηλικιακή ομάδα 15-24 υποδεικνύει ότι η το να είναι κανείς εκατομμυριούχος είναι λιγότερο συνηθισμένο στις νεότερες ομάδες.

5 Σε ποια χώρα να ζω τέλος πάντων για να γίνω πλούσιος;

5.1 Ψευδοκώδικας

Ο σκοπός της μεθόδου Map είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, εξάγοντας τα χαρακτηριστικά "country", "cpi" και " finalWorth", και να εκπέμπει ζεύγη key-value. Κάθε ζεύγος έχει ως key το ζευγάρι/tuple "country-cpi" (υλοποιούμενο με κώδικα Java όπως παρουσιάζεται στο 6.3), καθώς σκοπός είναι να γκρουπ-άρουμε όλες οι εγγραφές με κοινή χώρα, ώστε στο επόμενο βήμα να επεξεργαστούμε μαζί. Επίσης, στο τελικό αποτέλεσμα χρειαζόμαστε το cpi της κάθε χώρας, οπότε πρέπει να το εκπέμπουμε από την Map, και καθώς το cpi μιας χώρας είναι μοναδικό για αυτήν, και ανεξάρτητο από την κάθε εγγραφή (κα άρα δεν επηρεάζει το γκρουπ-άρισμα βάσει χώρας), το εκπέμπουμε ως μέρος του key, που δεν θα επεξεργαστούμε μελλοντικά. Ως value, εκπέμπεται η αντίστοιχη τιμή του "finalWorth" από τη συγκεκριμένη εγγραφή.

Αλγόριθμος 7: Ψευδοκώδικας για Mapping()

```
class Mapper
    method Map (docID d, null)
        foreach line I in d
            emit (pair(country c, cpi_country cpi),
            finalWorth w)
```

Η μέθοδος Reduce διαχειρίζεται το key "country-cpi" και την αντίστοιχη λίστα των τιμών "finalWorth" που έχει προκύψει. Υπολογίζει το συνολικό άθροισμα των τιμών " finalWorth" που αντιστοιχούν σε αυτήν τη χώρα. Στη συνέχεια, εκπέμπει το key "country", το key "cpi", το άθροισμα των τιμών "finalWorth" και το μήκος της λίστας με τα finalWorth (το οποίο αναπαριστά το πόσες εγγραφές/εκατομμυριούχοι υπάρχουν στη λίστα της χώρας). Αυτή η διαδικασία επιτρέπει τον υπολογισμό του συνολικού πλούτου και το πλήθος των εκατομμυριούχων ανά χώρα.

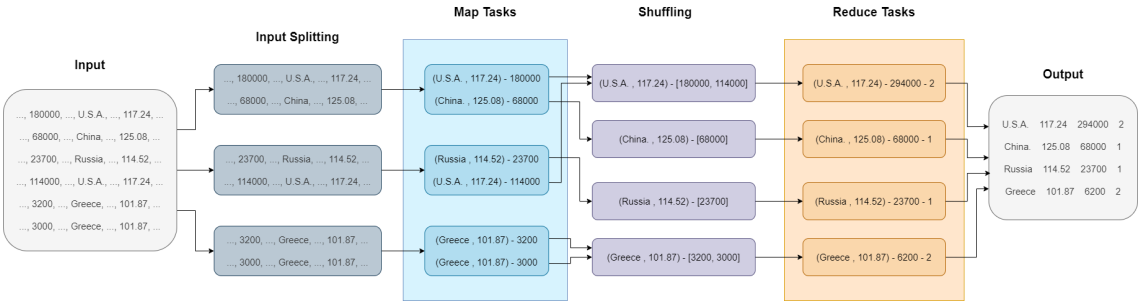
Αλγόριθμος 8: Ψευδοκώδικας για Reducing()

```
class Reducer
    method Reduce (pair (country c, cpi_country cpi),
```

```
    list worths[])
        sum = 0
        foreach w in worths[]
            sum = sum + w
        emit (country c, cpi_country cpi, sum,
        worths.length)
```

5.2 Σχηματική εκτέλεση

Για να κάνουμε τα MapReduce tasks και τα αποτελέσματα τους πιο εύκολα στην κατανόηση, υλοποιούμε ένα απλό και σύντομο παράδειγμα, σε μορφή διαγράμματος, πάνω σε περιορισμένες εγγραφές. Ώστε να καλύψουμε τις περισσότερες τις περιπτώσεις και να μια ρεαλιστική εικόνα/μικρογραφία του συστήματος μας, έχουμε συμπεριλάβει την ίδια τιμή σε 2 εγγραφές που βρίσκονται και εντός του ίδιου μπλοκ και σε διαφορετικά μπλοκς δεδομένων. Το παράκατω διάγραμμα μπορείτε να το βρείτε στο Examples_Schemas/q3schema.png.



Εικόνα 7: Παράδειγμα και Σχηματική υλοποίηση της διαδικασίας MapReduce για την εύρεση των στατιστικών κάθε χώρας

5.3 Αποτελέσματα

Τα αποτελέσματα που έχουμε ως file αρχείο, τρέχοντας τον αντίστοιχο κώδικα Java στο HDFS, αναπαριστόνται παράκατω σε μορφή πίνακα (για λόγους καλύτερης οργάνωσης). Το file αρχείο υπάρχει στον φάκελο του πρότζεκτ υπό το όνομα "Q3". Παρακάτω παραθέτονται μόνο τα πρώτα 20 αποτελέσματα ενδεικτικά, ενώ τα πλήρη αποτελέσματα είναι εντός του Result_files/Q3.file.

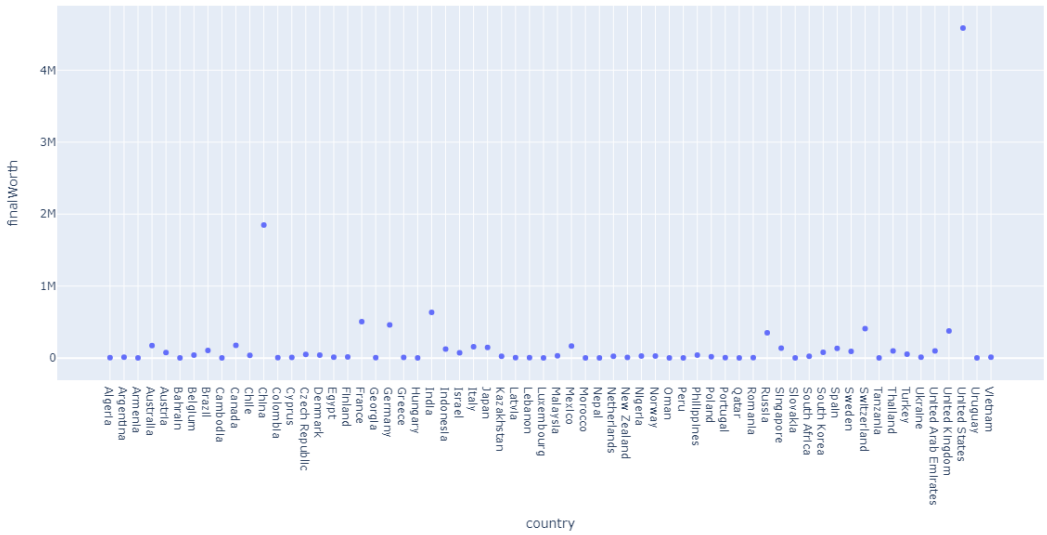
Πίνακας 4: Στατιστικά χωρών (Top 20)

Χώρα	**CPI**	**Άθροισμα πλούτου**	**# εκατομμυριούχων**
Algeria	151.36	4600	1
Argentina	232.75	11000	4
Armenia	129.18	1200	1
Australia	119.8	175000	44
Austria	118.06	75400	11
Bahrain	117.59	1500	1
Belgium	117.11	41200	3
Brazil	167.4	104800	44
Cambodia	127.63	2800	1
Canada	116.76	175300	43
Chile	131.91	36400	6
China	125.08	1847600	529
Colombia	140.95	6400	1
Cyprus	102.51	9600	5
Czech Republic	116.48	50800	8
Denmark	110.35	40900	7
Egypt	288.57	13500	4
Finland	112.33	14000	7
France	110.05	505900	36
Georgia	133.61	4900	1

5.4 Γραφική απεικόνιση και Συμπεράσματα

Προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα scatter plot, μεταξύ των χωρών και το άθροισμα του πλούτου σε κάθε μια από αυτές. Μέσω του scatter plot θέλουμε να παρατηρήσουμε πιθανά μοτίβα στα δεδομένα όταν τα αναπαριστούμε.

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε κάποια συμπεράσματα ως προς το ποσό πλούτο που συσσωρεύεται σε κάθε χωρά. Οι Ηνωμένες Πολιτείες ξεχωρίζουν με τη υψηλότερη συνολική περιουσία, αντανακλώντας την κυρίαρχη οικονομική τους θέση παγκοσμίως. Η Κίνα ακολουθεί πολύ κοντά, δείχνοντας την έντονη οικονομική της ανάπτυξη. Ευρωπαϊκές χώρες όπως η Ελβετία, η Γερμανία και το Ηνωμένο Βασίλειο επίσης εκδηλώνουν σημαντικές συγκεντρώσεις πλούτου. Ωστόσο, η αξιολόγηση της



Εικόνα 8: Scatter plot χώρας-ατόμων

επίδρασης μιας χώρας στο ατομικό πλούτο των εκατομμυριούχων απαιτεί να ληφθεί υπόψη και το γενικός πλούτος της ίδιας της χώρας, καθώς δεν μπορούμε να βγάλουμε κάποιο ιδιαίτερο συμπέρασμα ή συσχέτιση του πλούτου και τα χωράς από αυτά τα δεδομένα. Τα δεδομένα υποδηλώνουν ένα μεγάλο φάσμα παραγόντων που επηρεάζουν τον πλούτο των εκατομμυριούχων, συμπεριλαμβανομένων των οικονομικών πολιτικών και της κατανομής του πλούτου εντός κάθε χώρας.

6 Με ποιο τομέα πρέπει να ασχοληθώ για να πιάσω την καλή;

6.1 Ψευδοκώδικας

Ο σκοπός της βοηθητικής μεθόδου Map1 είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, εξάγοντας το χαρακτηριστικό ”finalWorth” και εκπέμποντας ζεύγη key-value. Αυτά τα ζεύγη αποτελούνται από το key ”total worth” ως απλό string (καθώς θέλουμε να γκρουπ-άρουμε όλες τις ηλικίες μαζί με κοινό key, ώστε στο επόμενο βήμα να τις συγκρίνουμε όλες μεταξύ τους) και ως value την αντίστοιχη τιμή του χαρακτηριστικού στην εγγραφή. Η βοηθητική μέθοδος Reduce1 χειρίζεται το key ” total worth” και την αντίστοιχη λίστα με τιμές finalWorth που έχει προκύψει. Υπολογίζει το συνολικό άθροισμα αυτών των τιμών και το εκπέμπει. Αυτή η διαδικασία επιτρέπει τον υπολ-

ογισμό του συνολικού πλούτου από όλες τις εγγραφές, η οποία χρησιμοποιείται στο επόμενο στάδιο της επεξεργασίας των δεδομένων.

Αλγόριθμος 9: Ψευδοκώδικας για βοηθητικό Mapping()

```
class Mapper1
    method Map1 (docID d, null)
        foreach line I in d
            emit (key "total_worth", finalWorth w)
```

Αλγόριθμος 10: Ψευδοκώδικας για βοηθητικό Reducing()

```
class Reducer1
    method Reduce1 (key "total_worth", list worths[])
        sum = 0
        foreach w in worths[]
            sum = sum + w
        emit ("total_worth", sum)
```

Ο σκοπός της μεθόδου Map2 είναι να «διαβάσει» κάθε γραμμή του εγγράφου μας, εξάγοντας τα χαρακτηριστικά "category", "finalWorth" και "personName", και να εκπέμπει ζεύγη key-value. Κάθε ζεύγος έχει ως key την τιμή του "category" στη συγκεκριμένη εγγραφή, διότι στη συνέχεια θέλουμε να γκρουπ-άρουμε όλες τις εγγραφές με κοινό key (δηλαδή κοινή κατηγορία), προκειμένου να τις επεξεργαστούμε μαζί. Ως value, εκπέμπεται το ζευγάρι/tuple " finalWorth-personName", καθώς για κάθε κατηγορία, θέλουμε να επεξεργαστούμε τις τιμές του finalWorth των εκατομμυριούχων της, έχοντας γνώση πάνω σε ποιον εκατομμυριούχο δουλεύουμε κάθε φορά.

Αλγόριθμος 11: Ψευδοκώδικας για Mapping()

```
class Mapper2
    method Map2 (docID d, null)
        foreach line I in d
            emit (category c,
            pair(finalWorth w, personName n))
```

Η μέθοδος Reduce2 διαχειρίζεται το key "category" και την αντίστοιχη λίστα των τιμών/ζευγαριών "finalWorth-personName" που έχει προκύψει. Υπολογίζει το συνολικό άθροισμα των τιμών "finalWorth" που αντιστοιχούν σε αυτήν τη κατηγορία, και τις συγκρίνει προκειμένου να βρει το max, και έπειτα κρατά το όνομα "personName" που αντιστοιχεί σε αυτή τη τιμή. Στη συνέχεια, εκπέμπει το key "category", το άθροισμα των τιμών "finalWorth", το ποσοστό επί του συνολικού πλούτου που κατέχει η κατηγορία αυτή (υπολογίζοντας το με τη βοήθεια του συνολικού πλούτο που ξέρουμε πλέον καθώς βρέθηκε από το προηγούμενο MapReduce), το μήκος της λίστας με τα ζευγάρια (το οποίο αναπαριστά το πόσες εγγραφές/εκατομμυριούχοι υπάρχουν στη λίστα της κατηγορίας), τον μέσο των τιμών "finalWorth" και το όνομα με τον περισσότερο πλούτο. Αυτή η διαδικασία επιτρέπει τον υπολογισμό στατιστικών του πλούτου ανά κατηγορία.

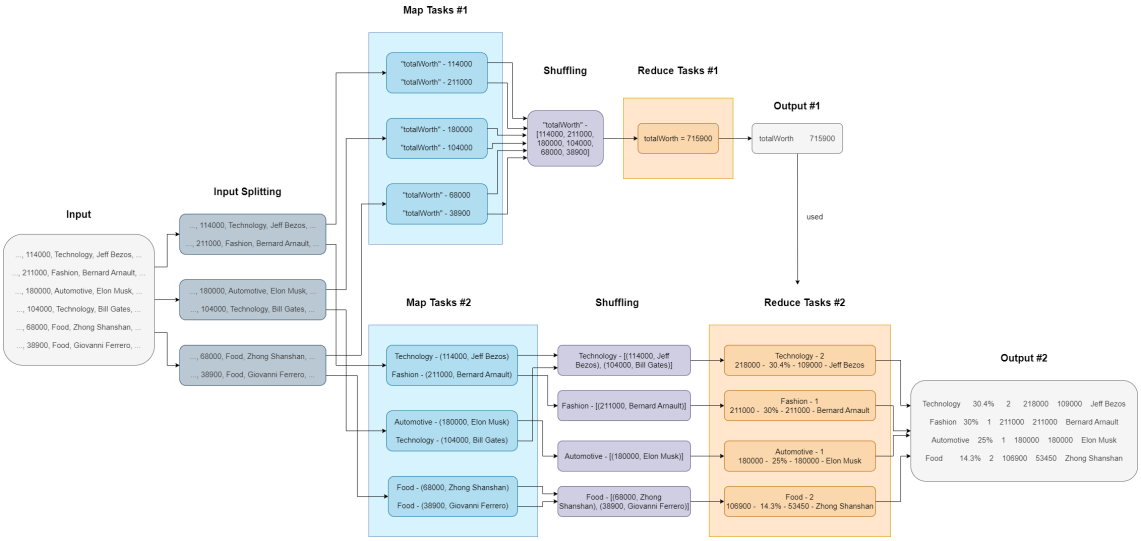
Αλγόριθμος 12: Ψευδοκώδικας για Reducing()

```
class Reducer2
    total_worth = {output of Reducer1}
    method Reduce2 (category c, list millioners[])
        sum = 0
        max = inf
        best = null
        length = millioners.length
        foreach m in millioners[]
            sum = sum + m.finalWorth
        if (m.finalWorth > max)
            best = m.personName
            max = m.finalWorth
        emit (category c, sum*100/total_worth, length,
        sum, sum/length, best)
```

6.2 Σχηματική εκτέλεση

Για να κάνουμε τα MapReduce tasks και τα αποτελέσματα τους πιο εύκολα στην κατανόηση, υλοποιούμε ένα απλό και σύντομο παράδειγμα, σε μορφή διαγράμματος, πάνω σε περιορισμένες εγγραφές. Ώστε να καλύψουμε τις περισσότερες τις περιπτώσεις και να μια ρεαλιστική εικόνα/μικρογραφία του συστήματος μας, έχουμε συμπερ-

ιλάβει την ίδια τιμή σε 2 εγγραφές που βρίσκονται και εντός του ίδιου μπλοκ και σε διαφορετικά μπλοκς δεδομένων. Το παράκατω διάγραμμα μπορείτε να το βρείτε στο Examples_Schemas/q4schema.png.



Εικόνα 9: Παράδειγμα και Σχηματική υλοποίηση της διαδικασίας MapReduce για την εύρεση των στατιστικών κάθε κατηγορίας

6.3 Σχολιασμός κώδικα Java

Αρχικά, πρέπει να σημειώσουμε την ανάγκη για δύο προγράμματα MapReduce, άρα και δύο ξεχωριστά JAR αρχεία. Προκειμένου να υπολογίσουμε το ποσοστό του συνολικού πλούτου που κατέχει κάθε κατηγορία, πρέπει πρώτα να υπολογίσουμε το συνολικό πλούτο. Αυτό δεν μπορεί να γίνει στο ίδιο MapReduce πρόγραμμα, διότι τα reduce tasks θα είναι χωρισμένα ανά κατηγορία και δεν θα υπάρχει καθολική/global γνώση του συνολικού πλούτου στο σύστημα. Έτσι, δημιουργήσαμε ένα απλό βοηθητικό πρόγραμμα MapReduce για να υπολογίσουμε το συνολικό πλούτο (q4.jar). Στη συνέχεια, μόλις λάβαμε το αποτέλεσμά από το βοηθητικό πρόγραμμα, το ενσωματώσαμε hard-coded στο πρόγραμμα MapReduce που κάνει τον του υπολογισμού των στατιστικών κάθε κατηγορίας (q4b.jar).

Επιπλέον, για τη δημιουργία των ζευγαριών/tuples, δημιουργήσαμε ένα νέο αντικείμενο/κλάση τύπου MillionaireData.java. Αυτό το αντικείμενο έχει ως πεδία τον πλούτο finalWorth και το όνομα του ατόμου personName που χρειαζόμασταν, και περιλαμβάνει μεθόδους για τον καθορισμό/setting και την ανάκτηση/getting των τιμών

αυτών των πεδίων. Ουσιαστικά, δεν εκπέμπουμε ή επεξεργαζόμαστε ζευγάρια ή tuples, αλλά αντικείμενα τύπου MillionaireData.

6.4 Αποτελέσματα

Τα αποτελέσματα που έχουμε ως file αρχείο, τρέχοντας τον αντίστοιχο κώδικα Java στο HDFS, αναπαριστόνται παράκατω σε μορφή πίνακα (για λόγους καλύτερης οργάνωσης). Το file αρχείο υπάρχει στον φάκελο του πρότζεκτ υπό το όνομα "Result_files/Q4b".

Πίνακας 5: Στατιστικά κατηγοριών

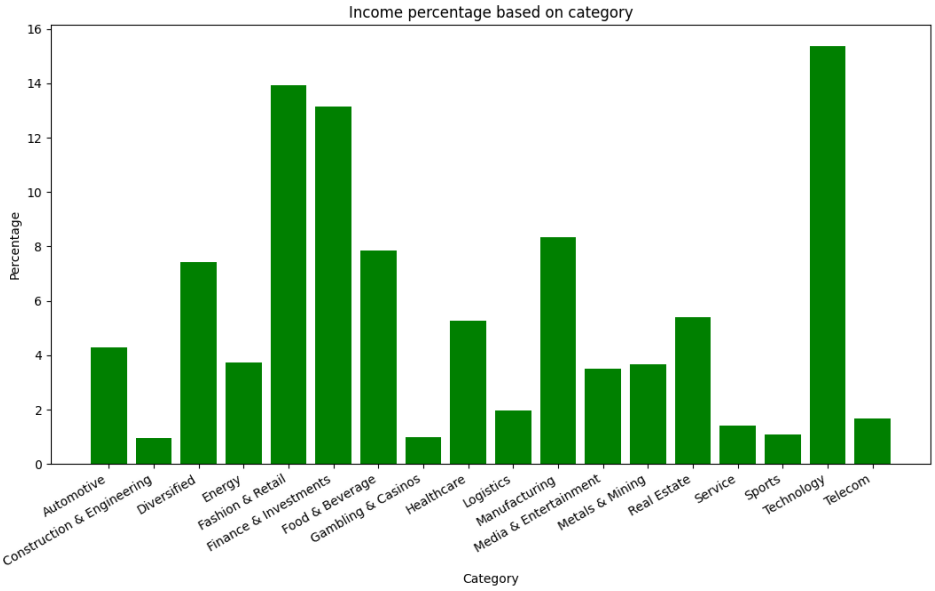
Κατηγορία	**%**	**#**	**Sum**	**M.O.**	**Top**
Automotive	4.30%	73	525300	7195	Nirmal Minda
Construction & Engineering	0.97%	45	118500	2633	Riley Bechtel & family
Diversified	7.42%	187	905200	4840	Zadik Bino & family
Energy	3.72%	100	453500	4535	Arthur Irving
Fashion & Retail	13.92%	266	1698800	6386	Masateru Uno & family
Finance & Investments	13.15%	372	1605100	4314	Warren Buffett
Food & Beverage	7.84%	212	957200	4515	Amit Burman
Gambling & Casinos	0.99%	25	120500	4820	Jon Yarbrough
Healthcare	5.27%	201	643200	3200	Vadim Yakunin
Logistics	1.96%	40	239500	5987	Rafaela Aponte-Diamant
Manufacturing	8.35%	324	1019000	3145	Igor Rybakov
Media & Entertainment	3.50%	91	427500	4697	Jason Jiang
Metals & Mining	3.66%	74	446800	6037	John Hancock
Real Estate	5.39%	193	657400	3406	Angela Leong
Service	1.42%	53	173400	3271	Tengyun Nie & family
Sports	1.10%	39	134500	3448	Jeffrey Lurie & family
Technology	15.38%	314	1877900	5980	Eric Ya Shen
Telecom	1.67%	31	203500	6564	Rocco Commisso

6.5 Γραφικές απεικόνισεις και Συμπεράσματα

Bar graph

Στη πρώτη περίπτωση, προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα bar graph, μεταξύ των κατηγοριών και του ποσοστού του συνολικού πλούτου που κατέχουν. Η επιλογή bar chart έγινε καθώς οι κατηγορίες στον

οριζόντιο άξονα αντιπροσωπεύουν ένα nominal χαρακτηριστικό, που σημαίνει ότι δεν υπάρχει κάποια εγγενής σειρά ή κλίμακα μεταξύ τους (σε αντίθεση με του αριθμούς σε ένα numeric χαρακτηριστικό που αυξάνονται όσο προχωράμε στον άξονα).



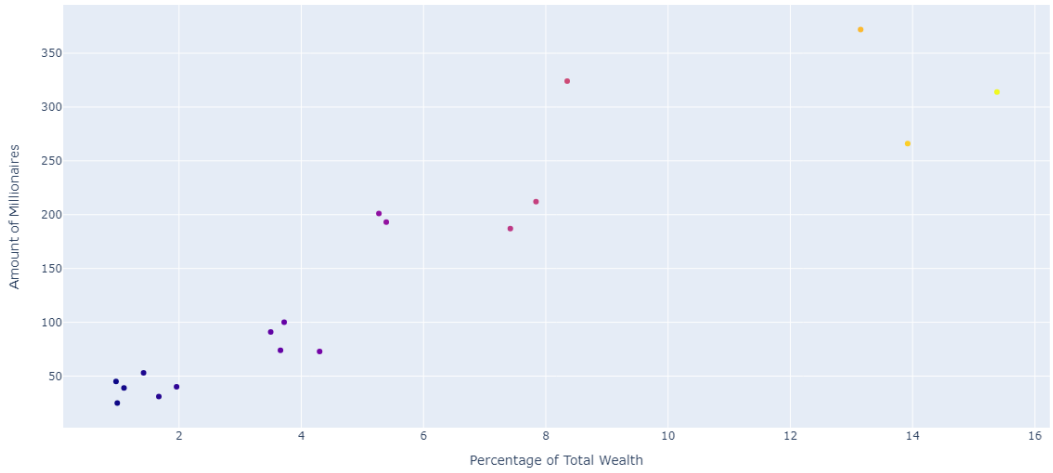
Εικόνα 10: Histogrram κατηγορίας-ποσοστού πλούτου

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε συμπεράσματα ως προς το ποσό του πλούτου που συσσωρεύει η κάθε κατηγορία. Η Τεχνολογία (Technology) εμφανίζεται ως η κυρίαρχη κατηγορία, κατέχοντας το υψηλότερο ποσοστό του συνολικού πλούτου, ακολουθούμενη στενά από τη Μόδα (Fashion Retail) και την Χρηματοοικονομία (Finance Investments), αναμενόμενο αποτέλεσμα καθώς αυτοί θεωρούνται από τος πιο κοινούς και προσοδοφόρους τομείς για εκατομμυριούχους. Ορισμένες κατηγορίες, όπως οι Construction Engineering, Gambling Casinos, Logistics, Service και Sports, έχουν σχετικά χαμηλά ποσοστά, υποδεικνύοντας μικρή συμμετοχή στη συνολική κατανομή πλούτου.

Scatter plot

Στη δεύτερη περίπτωση, προκειμένου να αναπαραστήσουμε τα αποτελέσματα που λάβαμε χρησιμοποιούμε ένα scatter plot, μεταξύ του αριθμού των εκατομμυριούχων σε κάθε κατηγορία και του ποσοστού του συνολικού πλούτου που κατέχουν. Μέσω του scatter plot, όπου κάθε σημείο αναπαριστά μια κατηγορία, θέλουμε να παρατηρήσουμε πιθανά

μοτίβα στη κατανομή του πλούτου στις κατηγορίες, αλλά και ταυτόχρονα τον αριθμό ατόμων σε αυτή.



Εικόνα 11: Scatter plot ποσοστού πλούτου-ατόμων

Από το παραπάνω διάγραμμα μπορούμε να βγάλουμε ίσως κάποια πιθανή συσχέτιση μεταξύ του ποσοστού του συνολικού πλούτου που κατέχουν οι εκατομμυριούχοι σε κάθε κατηγορία και τον αριθμό των εκατομμυριούχων μέσα σε αυτές τις κατηγορίες. Η Τεχνολογία, η Μόδα και τα Χρηματοοικονομικά ξεχωρίζουν ως κατηγορίες με υψηλή συγκέντρωση πλούτου και σημαντικό αριθμό εκατομμυριούχων, σε επίπεδο ακραίων τιμών. Παρόλα αυτά παρατηρούμε παραδείγματος χάριν, πως ενώ τα Χρηματοοικονομικά απασχολούν μεγαλύτερο αριθμό εκατομμυριούχων από κάθε άλλη κατηγορία, η Μόδα και η Τεχνολογία έχουν συσσωρεύσει μεγαλύτερο πλούτο στην αγορά. Δηλαδή, βλέπουμε ότι ορισμένοι κλάδοι εμφανίζουν υψηλότερο πλούτο ανά άτομο, και άλλοι δείχνουν πως έχουν λιγότερο πλούτο ανά άτομο με μεγαλύτερη όμως συμμετοχή. Το διάγραμμα παρουσιάζει μια θετική συσχέτιση (positive correlation), καθώς προφανώς ο πλούτος αυξάνεται, όσο αυξάνεται και ο αριθμός εκατομμυριούχων σε μια κατηγορία.