ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Project 1

Εργασία μαθήματος ΔΙΑΧΕΙΡΗΣΗ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ

Συγγραφέας:
Γυφτονικολός Νικόλαος, dit18039@go.uop.gr
Λέκκας Γεώργιος, dit18108@go.uop.gr
Πετσέλης Ιωάννης, dit17158@go.uop.gr

Δεκέμβριος 2022

Πίνακας Περιεχομένων

1 Εγκατάσταση Hadoop σε περιβάλλον Ubuntu

Για την υλοποίηση της πρώτης εργασίας μεταχειριστήκαμε Virtual Machine για την εγκατάσταση των Ubuntu στους υπολογιστές μας. Πιο συγκεκριμένα, χρησιμοποιήσαμε την έκδοση 3.2.4 του Hadoop Apache, για την οποία βρήκαμε και tutorial εγκατάστασης στο διαδίκτυο, του οποίου σύνδεσμο συμπεριλαμβάνουμε παρακάτω. Επιπροσθέτως, για την ομαλή λειτουργία του Hadoop με το Eclipse κάναμε χρήση κάποιο επιπλέον βιβλιοθηκών του Hadoop, τις οποίες επίσης αναφέρουμε παρακάτω.

Ο σύνδεσμος για την εγκατάσταση του Hadoop σε περιβάλλον Ubuntu :

https://phoenixnap.com/kb/install-hadoop-ubuntu?fbclid=IwAR2_xQKmynQ-HFoXyeB067CXHcqq4aWpRov2aHO0Hl3XDqLJankgDGYgKpM

Οι βιβλιοθήκες που προσθέσαμε στο Eclipse :

- commons-cli-1.2.jar
- hadoop-common-3.2.4.jar
- hadoop-mapreduce-client-core-3.2.4.jar

2 Λειτουργικότητα του συστήματος

Για τις ανάγκες υλοποίησης του τελευταίου ερωτήματος χρειάστηκε να μειώσουμε το μέγεθος των εγγραφών του αρχείου του click_stream.csv, διότι τα Virtual Machine που δημιουργήσαμε δε μπορούσαν να επεξεργαστούν τα δεδομένα και να μας παρουσιάσουν ένα τελικό αποτέλεσμα. Το νεό αρχείο που δημιουργήσαμε ονομάζεται test.csv και βρίσκεται στα παραδοτέα.

3 Μελέτη των δεδομένων

customer.csv :

- customer_id : nominal
- first_name : nominal
- last_name : nominal
- username : nominal
- email : nominal
- gender : binary
- birthdate : nominal
- device_type : binary
- device_id : nominal
- device_version : nominal
- home_location_lat : numeric - interval
- home_location_long : numeric - interval
- home_location : nominal
- home_country : nominal
- first_join_date : nominal

product.csv :

- id : nominal
- gender : binary
- masterCategory : nominal
- subCategory : nominal
- articleType : nominal
- baseColour : nominal
- season : nominal
- year : numeric
- usage : nominal
- productDisplayName : nominal

transactions.csv :

- created_at : numeric - interval
- customer_id : nominal
- booking_id : nominal
- session_id : nominal
- product_metadata
  - product_id : nominal
  - quantity : numeric - ratio
  - item_price : numeric - ratio
- payment_method : nominal
- payment_status : binary
- promo_amount : numeric - ratio
- promo_code : nominal
- shipment_fee : numeric - interval
- shipment_date_limit : numeric - interval
- shipment_location_lat : numeric - interval
- shipment_location_long : numeric - interval
- total_amount : numeric - ratio

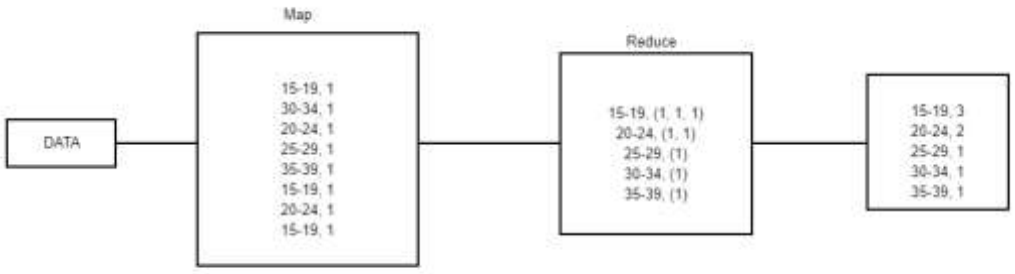
click_stream.csv :

- session_id : nominal
- event_name : nominal
- event_time : numeric - interval
- event_id : nominal
- traffic_source : binary
- event_metadata
  - product_id : nominal
  - quantity : numeric - ratio
  - item_price : numeric – ratio
  - promo_code : nominal
  - promo_amount : numeric - ratio
  - payment_status : binary
  - search_keywords : nominal

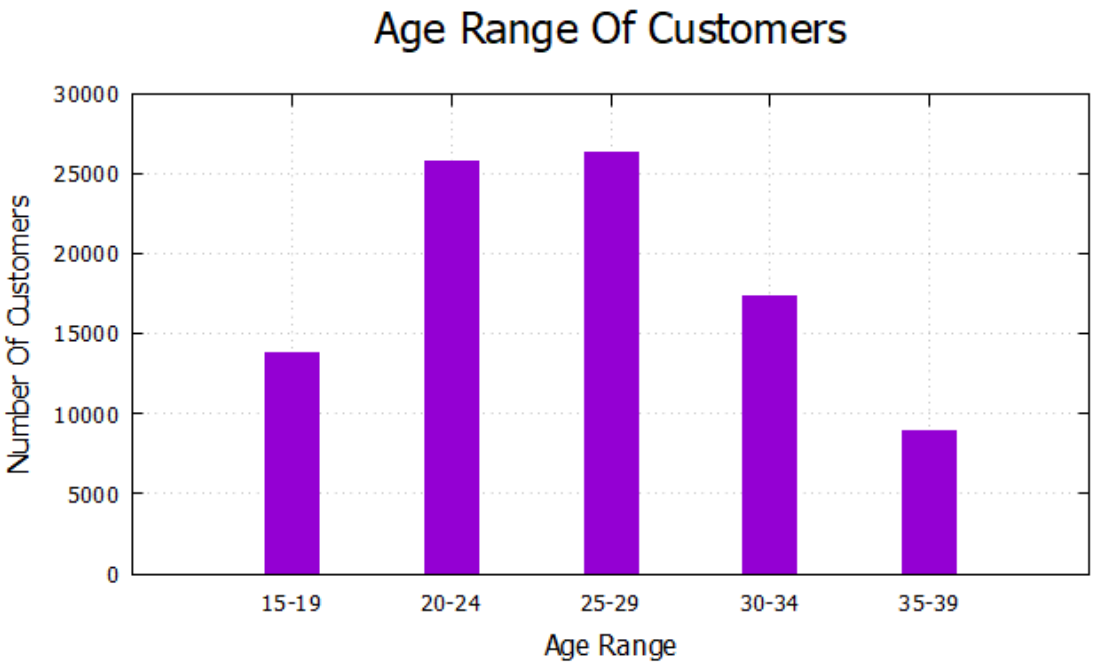
4 Ο ψευδοκώδικας, μαζί με ένα παράδειγμα και μια σχηματική εκτέλεση για κάθε ένα από τα ερωτήματα

4.1 Ηλικιακό φάσμα πελατών

```
Age_Group {

        void Map (key, String data values) {
                if ( birthdate <= 2007 AND birthdate > 2003 )
                        key = " 15 – 19  ";
                if ( birthdate <= 2002 AND birthdate > 1998 )
                        key = " 20 – 24  ";
                if ( birthdate <= 1997 AND birthdate > 1993 )
                        key = " 25 – 29  ";
                if ( birthdate <= 1992 AND birthdate > 1988 )
                        key = " 30 – 34  ";
                if ( birthdate <= 1987 AND birthdate > 1983 )
                        key = " 35 – 39  ";

                emit(key, 1);
        }

        void Reduce (Text key, int [] counts) {
                int sum = 0;
                foreach (int i in counts)
                        sum += i;

                emit(key, sum);
        }
}
```



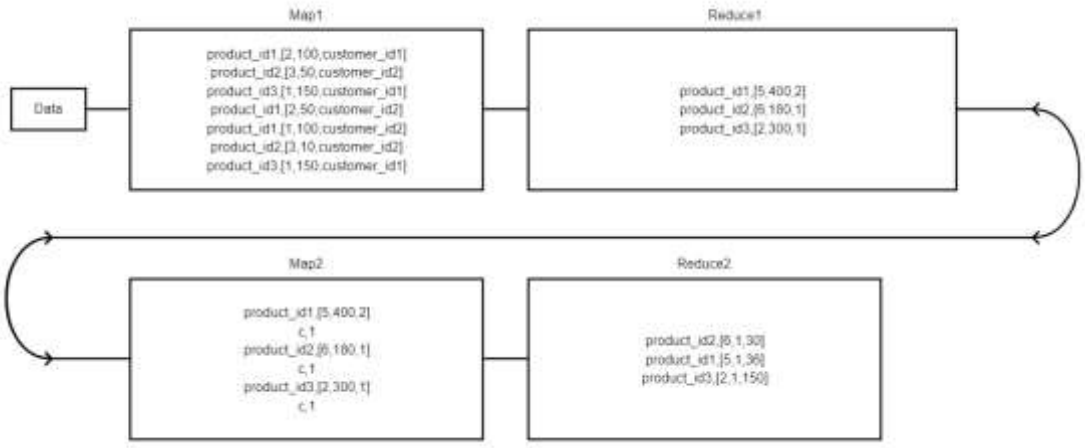
15-19	13793
20-24	25751
25-29	26335
30-34	17393
35-39	8947



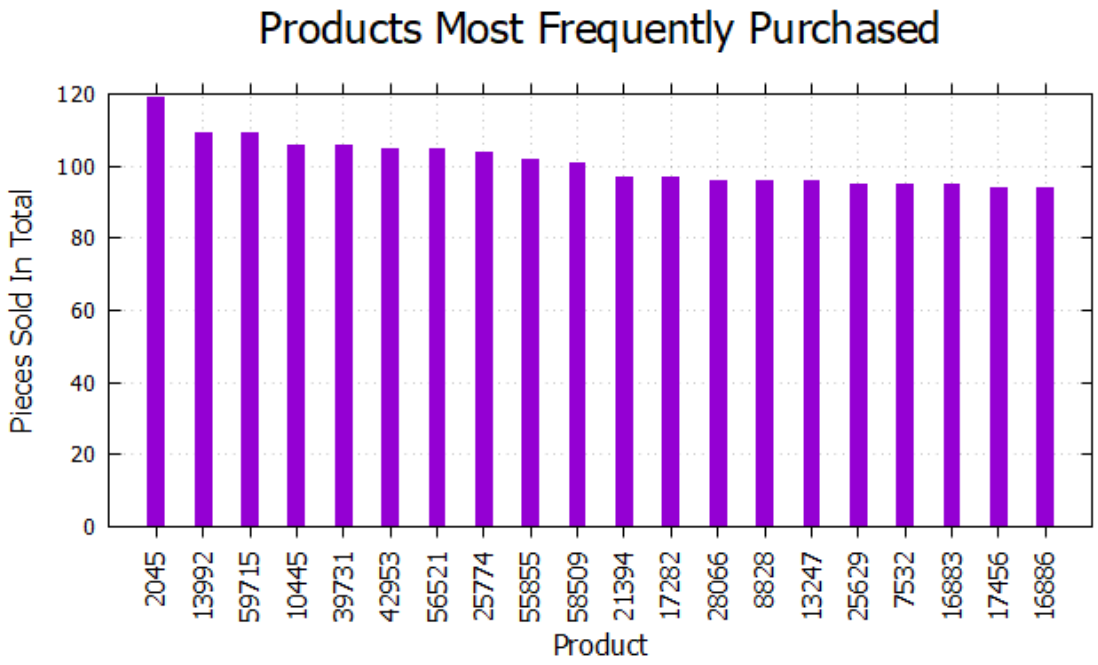
4.2 Ανάδειξη των προϊόντων που αγοράζονται πιο συχνά από τους πελάτες του καταστήματος

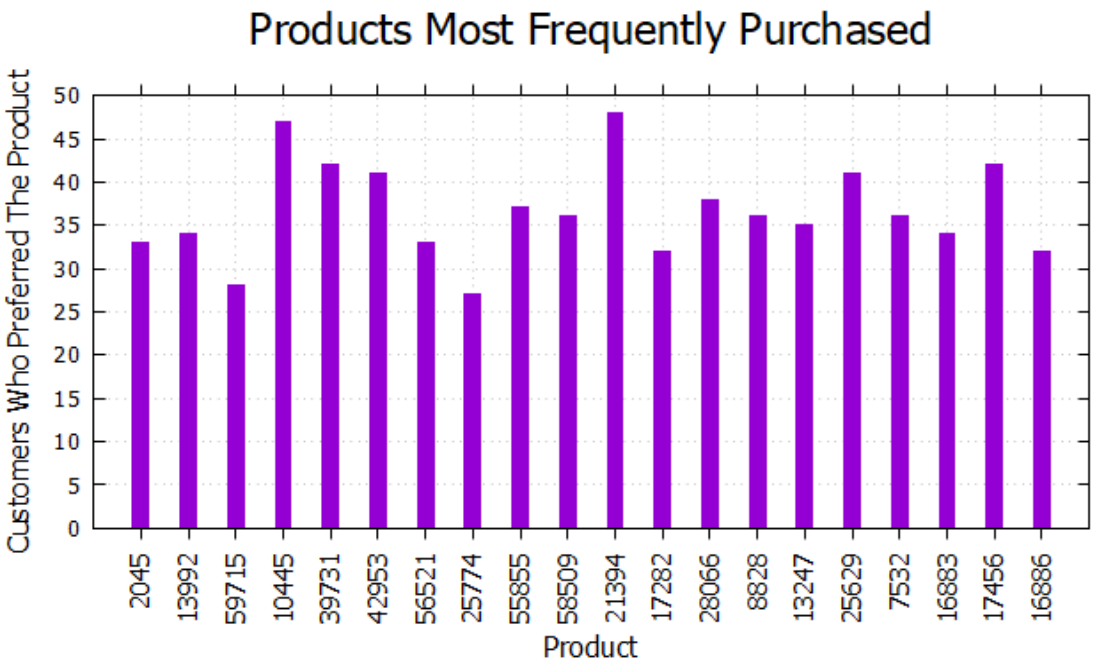
```
Ranking_Products {

    void map1 (key, String values) {
        key.set(values.product_id)
        value = values.quantity + "," + values.item_price + "," + values.customer_id;
        emit(key,value);
    }

    void reduce1 (key, String[] values) {
        foreach (String s in values){
            sum_quantity += s.quantity;
            sum_price += s.quantity*item_price;
            if(customer_id.exists(list)==false)
                add.list(s.customer_id);
        }
        value = sum_quantity + "," + sum_price + "," + list.size();
        emit(key, value);
    }

    void map2 (key, String values) {
        emit(key, values);
        emit(c,1);
    }
```

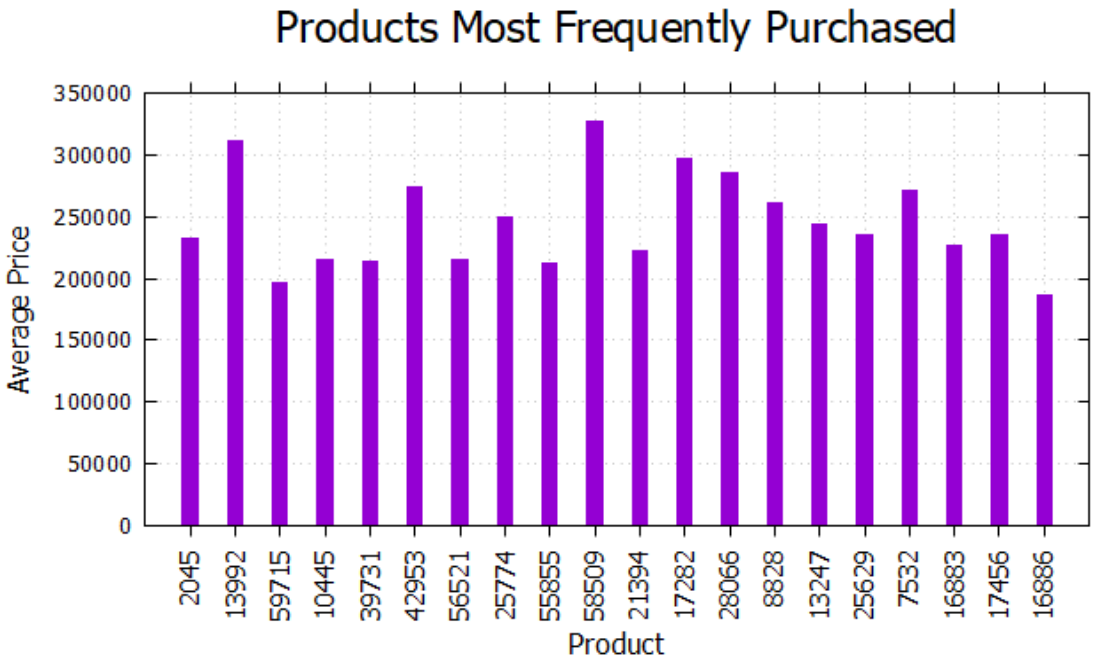
```
        void reduce2 (key, String[] values) {
                int i = 0, c = 0;
                foreach(String s in values){
                if(key==c)
                        c = s;
                else
                        add.list(key, s.quantity, s.price/s.quantity,
s.number_of_customers);
                }
                if(c!=0){
                        sort.list();
                        set.key(++i);
                        value = list.quantity + " " + list.av_price + " " +
list.number_of_customers;
                        emit(key, value);
                }
        }
}
```



```
1 2045 119 33 232599.86
2 13992 109 34 310878.94
3 59715 109 28 196757.94
4 10445 106 47 214933.42
5 39731 106 42 213920.42
6 42953 105 41 273756.38
7 56521 105 33 215541.94
8 25774 104 27 250348.19
9 55855 102 37 212621.8
10 58509 101 36 326797.28
11 21394 97 48 222112.75
12 17282 97 32 297221.9
13 28066 96 38 285831.12
14 8828 96 36 261422.95
15 13247 96 35 244133.62
16 25629 95 41 235596.86
17 7532 95 36 271896.84
18 16883 95 34 226582.11
19 17456 94 42 235521.31
20 16886 94 32 186155.28
```



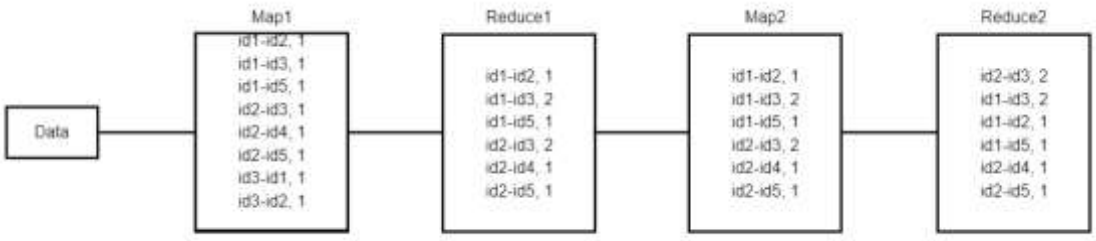

Products Most Frequently Purchased
Customers Who Preferred The Product
0
5
10
15
20
25
30
35
40
45
50
2045
13992
59715
10445
39731
42953
56521
25774
55855
58509
21394
17282
28066
8828
13247
25629
7532
16883
17456
16886
Product


Products Most Frequently Purchased
Average Price
0
50000
100000
150000
200000
250000
300000
350000
2045
13992
59715
10445
39731
42953
56521
25774
55855
58509
21394
17282
28066
8828
13247
25629
7532
16883
17456
16886
Product

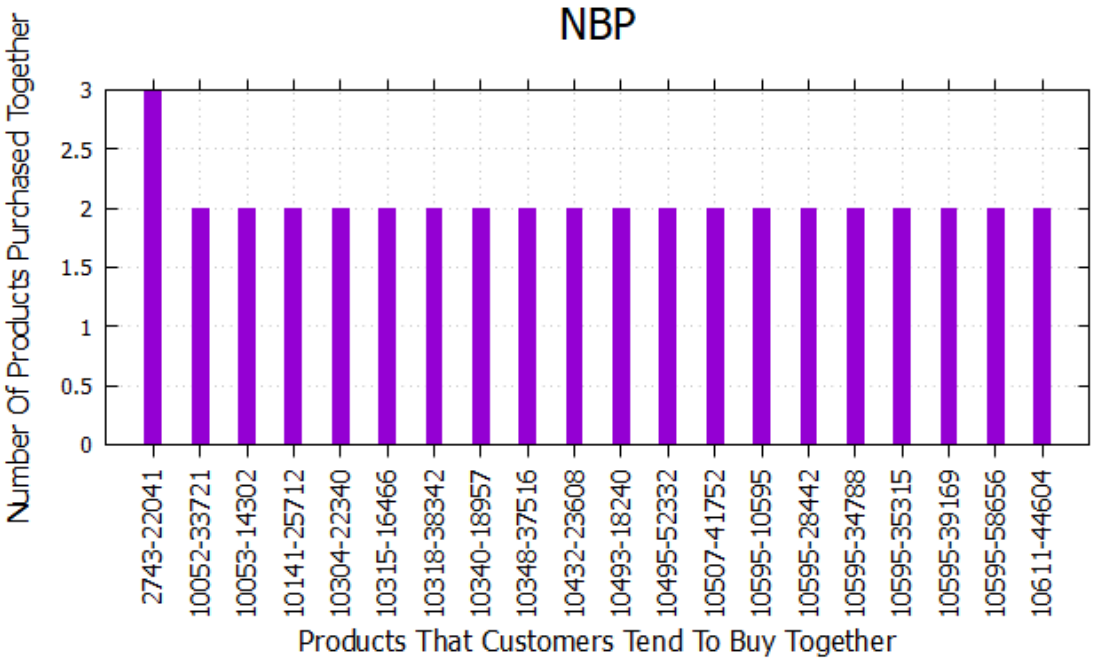
4.3 Ανάλυση του επόμενου καλύτερου προϊόντος (NBP)

```
Customer Behavior (){

    void map1 (key, String values) {
        int i = 0;
        String metadata = values.product_metadata;
        foreach(String s in metadata.product_id)
                add.list(s);
        for(int i = 0;i<list.size();i++)
                for(int j = i+1;j<list.size();j++) {
                        a = list.get(i);
                        b = list.get(j);
                        if(a<=b)
                                emit(a + "-" + b, 1);
                        if(a>=b)
                                emit((b + "-" + a, 1);
                        emit(c,1);
                }
          }

    }

    void reduce1 (key, String[] values) {
        int sum = 0;
        int count = 0;
        for (String s in values) {
                if(key!=c) {
                        sum += s;
                }else
                        count++;
          }
```

```
        if(sum>0)
                emit(key, sum);
        if(count>0)
                emit(c, count);
    }

    void map2 (key, String values) {
        emit(key, values);
    }

    void reduce2 (key, String[] values) {
            int i = 0, c = 0;
            foreach(String s in values){
                if(key==c)
                    c = s;
                else
                    add.list(key, s.number);
            }
            if(c!=0){
                sort.list();
                foreach(String str in list)
                emit(str.key, str.number);
            }
    }

}
```



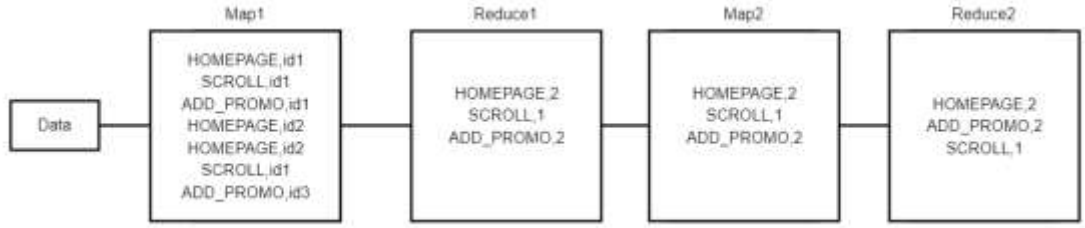
2743	22041	3
10052	33721	2
10053	14302	2
10141	25712	2
10304	22340	2
10315	16466	2
10318	38342	2
10340	18957	2
10348	37516	2
10432	23608	2
10493	18240	2
10495	52332	2
10507	41752	2
10595	10595	2
10595	28442	2
10595	34788	2
10595	35315	2
10595	39169	2
10595	58656	2
10611	44604	2



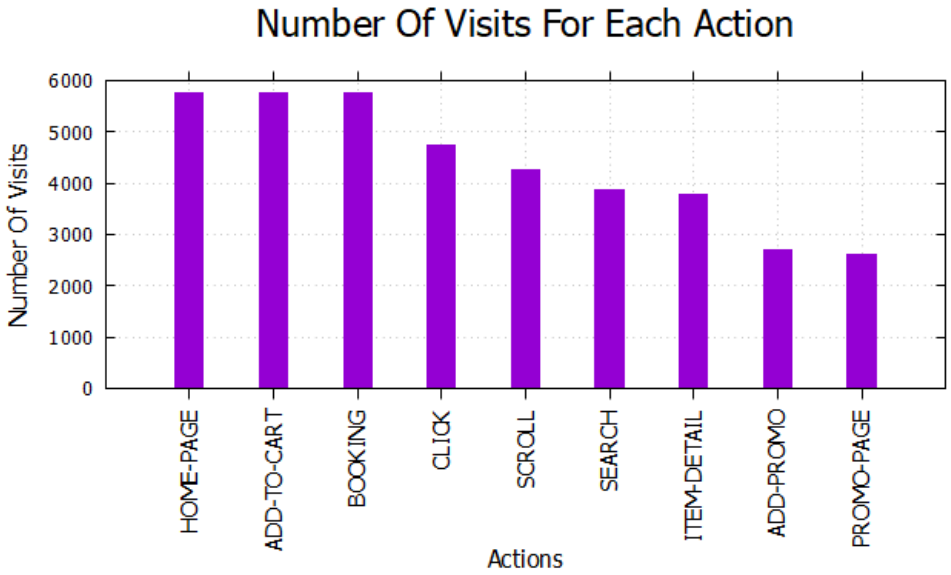
4.4 Κατανόηση της συνολικής αγοραστικής συμπεριφοράς των πελατών

```
Customer Behavior (){

    void map1 (key, String values) {
        key.set(values.event_name)
        emit(key, values.session_id);
    }

    void reduce1 (key, String[] values) {
        foreach(String s in values)
            if(s.session_id.exists(list)==false)
                add.list(s.session_id);
                value = list.size();
                emit(key, value);
    }

    void map2 (key, String values) {
        key.set(values.event_name)
        emit(key, values.numberOfIds);
        emit(c, 1);
    }

    void reduce2 (key, String[] values) {
        int c = 0;
        foreach(String s in values){
            if(key==c)
                c = s;
            else
                add.list(key, s.numberOfIds);
        }
```

```
            if(c!=0){
                    sort.list();
                    set.key(list.key);
                    value = list.numberOfIds;
                    emit(key, value);
            }
        }

}
```




```
HOMEPAGE        5776
ADD_TO_CART     5758
BOOKING 5756
CLICK   4734
SCROLL  4259
SEARCH  3864
ITEM_DETAIL     3780
ADD_PROMO       2692
PROMO_PAGE      2604
```


Number Of Visits For Each Action
Number Of Visits
6000
5000
4000
3000
2000
1000
0
HOME-PAGE
ADD-TO-CART
BOOKING
CLICK
SCROLL
SEARCH
ITEM-DETAIL
ADD-PROMO
PROMO-PAGE
Actions

5 Manual

Πρέπει να προσθέσουμε τις βιβλιοθήκες, όπως προαναφέραμε σε προηγούμενο βήμα. Στη συνέχεια, για την εκτέλεση του αρχείου πρέπει από το πρόγραμμα/editor (π.χ. Eclipse) να γίνει εξαγωγή του εκτελέσιμου αρχείου και στη συνέχεια να χρησιμοποιήσουμε την εντολή εκτέλεσης στο τερματικό στο σωστό directory (θα βάλουμε παραδείγματα στη συνέχεια).

./hadoop jar -path_to_jar_file Main -path_to_input -path_to_output

Η παρακάτω εντολή χρησιμοποιείται σε περίπτωση που έχουμε χρησιμοποιήσει 2 configurations κατά την υλοποίηση του ερωτήματος.
./hadoop jar -path_to_jar_file Main -path_to_input -path_to_output -path_to_output1