|  |  |
| --- | --- |
| Logo of NTUA | ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ |

Προχωρημένα Θέματα Βάσεων Δεδομένων

Εξαμηνιαία Εργασία

|  |
| --- |
| Ομάδα 88  Μάριος Κερασιώτης  03117890 |

Πίνακας Περιεχομένων

[1 Εισαγωγή 3](#_Toc126246118)

[2 Αρχικοποίηση και ρύθμιση συστήματος 3](#_Toc126246119)

[3 Εκτέλεση των ερωτημάτων 3](#_Toc126246120)

[4 Αποτελέσματα πειραμάτων 3](#_Toc126246121)

[4.1 Q1 3](#_Toc126246122)

[4.2 Q2 4](#_Toc126246123)

[4.3 Q3 5](#_Toc126246124)

[4.4 Q4 5](#_Toc126246125)

[4.5 Q5 6](#_Toc126246126)

[5 Χρόνοι εκτέλεσης 7](#_Toc126246127)

# Εισαγωγή

Σε αυτήν την εργασία καλούμαστε να επεξεργαστούμε δεδομένα όγκου, που αφορούν καταγραφές διαδρομών ταξί στην πόλη της Νέας Υόρκης. Συγκεκριμένα για να γίνει κάτι τέτοιο χρησιμοποιούμε τις τεχνολογίες Apache Hadoop 3.3.4 και Apache Spark 3.3.1. Το Apache Hadoop μας επιτρέπει να έχουμε έναν κατανεμημένο αποθηκευτικό χώρο στο σύνολο των virtual machines (VMs) μας και να επεξεργαζόμαστε τα δεδομένα με χρήση του μοντέλου Map Reduce. Το Apache Spark αποτελεί μια μηχανή ανάλυσης δεδομένων μεγάλου όγκου. Εμείς θα την χρησιμοποιήσουμε με τον python client της, το PySpark, σε περιβάλλον conda Python 3.10.

Τα δεδομένα πάρθηκαν από το Taxi & Limousine Commission (TLC) της Νέας Υόρκης. Συγκεκριμένα λήφθηκαν έξι αρχεία για κάθε μήνα από τον Ιανουάριο έως και τον Ιούνιο του 2022 και ένα αρχείο που αντιστοιχεί το πεδίο LocationID των παραπάνω αρχείων με τα προάστια και τις ζώνες της πόλης.

|  |
| --- |
| **Η υλοποίηση της εργασίας βρίσκεται στο GitHub στο** [**mariosker/advanced\_topics\_in\_database\_systems\_2022-ntua**](https://github.com/mariosker/advanced_topics_in_database_systems_2022-ntua) **repository, το οποίο είναι private. Έχει γίνει όμως πρόσκληση στον χρήστη dtsouma. Εάν δεν μπορείτε να μπείτε, εκτός από το να επικοινωνήσετε μαζί μου, μπορείτε να μπείτε και στο** [**gitfront**](https://gitfront.io/r/marioskeras/UmsEh3c9zZoh/advanced-topics-in-database-systems-2022-ntua/)**.** |

# Αρχικοποίηση και ρύθμιση συστήματος

Για να λειτουργήσει το σύστημα έπρεπε πρώτα να ρυθμιστούν τα hostnames στο /etc/hosts αρχείο. Αφότου γίνει αυτό κατεβάζουμε τα Hadoop και Spark. Στο Hadoop τροποποιούμε τα

* etc/hadoop/core-site.xml,
* etc/hadoop/hadoop-env.sh,
* etc/hadoop/hdfs-site.xml και
* etc/hadoop/workers

Για το spark αντίστοιχα τα αρχεία

* conf/spark-defaults.conf,
* conf/workers

Τέλος τροποποιούμε και το .bashrc.

# Εκτέλεση των ερωτημάτων

Το κάθε ερώτημα είναι αυτόνομο και δεν εξαρτάται από άλλα αρχεία. Έτσι σε ένα σύστημα με προ εγκατεστημένα και ρυθμισμένα τα Hadoop και Spark, και που έχει τα κατεβασμένα τα αρχεία από το TLC στον φάκελο raw\_data στο hdfs μπορούμε απλώς να τρέξουμε τα scripts με την εντολή $SPARK\_PATH/bin/spark-submit --master spark://master:7077 <script.py>.

# Αποτελέσματα πειραμάτων

Οι πίνακες Q1 και Q2 είναι οι ανάστροφοι των αρχικών αποτελεσμάτων για ευκολότερη ανάγνωση.

## Q1

Σκοπός του ερωτήματος αυτού είναι να βρεθεί η διαδρομή με το μεγαλύτερο φιλοδώρημα (tip) τον Μάρτιο και σημείο άφιξης το "Battery Park". Η υλοποίηση του πρέπει να γίνει με χρήση του DataFrame/ SQL API.

Το αποτέλεσμα του ερωτήματος είναι το εξής:

|  |  |
| --- | --- |
| VendorID | 2 |
| tpep\_pickup\_datetime | 2022-03-17 12:27:47 |
| tpep\_dropoff\_datetime | 2022-03-17 12:27:58 |
| passenger\_count | 1 |
| trip\_distance | 0 |
| RatecodeID | 1 |
| store\_and\_fwd\_flag | N |
| PULocationID | 12 |
| DOLocationID | 12 |
| payment\_type | 1 |
| fare\_amount | 2.5 |
| extra | 0 |
| mta\_tax | 0.5 |
| tip\_amount | 40 |
| tolls\_amount | 0 |
| improvement\_surcharge | 0.3 |
| total\_amount | 45.8 |
| congestion\_surcharge | 2.5 |
| airport\_fee | 0 |
| max(tip\_amount) | 40 |

## Q2

Σκοπός του ερωτήματος αυτού είναι να βρεθεί, για κάθε μήνα, η διαδρομή με το υψηλότερο ποσό στα διόδια, αγνοώντας τα μηδενικά ποσά. Η υλοποίηση του πρέπει να γίνει με χρήση του DataFrame/ SQL API.

Το αποτέλεσμα του ερωτήματος είναι το εξής:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VendorID | 1 | 1 | 1 | 1 | 1 | 1 |
| tpep\_pickup\_datetime | 2022-01-22 11:39:07 | 2022-02-18 02:33:30 | 2022-03-11 20:08:32 | 2022-04-29 04:31:21 | 2022-05-21 16:47:48 | 2022-06-12 16:51:46 |
| tpep\_dropoff\_datetime | 2022-01-22 12:31:09 | 2022-02-18 02:35:28 | 2022-03-11 20:09:45 | 2022-04-29 04:32:30 | 2022-05-21 17:05:47 | 2022-06-12 17:56:48 |
| passenger\_count | 1 | 1 | 1 | 2 | 1 | 9 |
| trip\_distance | 33.4 | 1.3 | 0 | 0 | 2.4 | 22 |
| RatecodeID | 1 | 1 | 1 | 1 | 3 | 1 |
| store\_and\_fwd\_flag | Y | N | N | N | N | N |
| PULocationID | 70 | 265 | 265 | 249 | 239 | 142 |
| DOLocationID | 265 | 265 | 265 | 249 | 246 | 132 |
| payment\_type | 4 | 1 | 1 | 3 | 3 | 2 |
| fare\_amount | 88 | 3 | 2.5 | 3 | 31.5 | 67.5 |
| extra | 0 | 0.5 | 1 | 3 | 0 | 2.5 |
| mta\_tax | 0.5 | 0.5 | 0.5 | 0.5 | 0 | 0.5 |
| tip\_amount | 0 | 19.85 | 48 | 0 | 0 | 0 |
| tolls\_amount | 193.3 | 95 | 235.7 | 911.87 | 813.75 | 800.09 |
| improvement\_surcharge | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| total\_amount | 282.1 | 119.15 | 288 | 918.67 | 845.55 | 870.89 |
| congestion\_surcharge | 0 | 0 | 0 | 2.5 | 0 | 2.5 |
| airport\_fee | 0 | 0 | 0 | 0 | 0 | 0 |

## Q3

Σκοπός του ερωτήματος αυτού είναι να βρεθεί, ανά 15 ημέρες, ο μέσος όρος της απόστασης και του κόστους για όλες τις διαδρομές με σημείο αναχώρησης διαφορετικό από το σημείο άφιξης. Η υλοποίηση του πρέπει να γίνει με χρήση του DataFrame/ SQL API και του RDD API.

Το αποτέλεσμα του ερωτήματος είναι το εξής:

|  |  |  |  |
| --- | --- | --- | --- |
| start | end | avg\_trip\_distance | avg\_total\_amount |
| 2021-12-29 02:00:00 | 2022-01-13 02:00:00 | 5.35568 | 20.2054 |
| 2022-01-13 02:00:00 | 2022-01-28 02:00:00 | 4.62712 | 18.9418 |
| 2022-01-28 02:00:00 | 2022-02-12 02:00:00 | 6.37884 | 19.5637 |
| 2022-02-12 02:00:00 | 2022-02-27 02:00:00 | 5.98213 | 19.9099 |
| 2022-02-27 02:00:00 | 2022-03-14 02:00:00 | 6.33147 | 20.6179 |
| 2022-03-14 02:00:00 | 2022-03-29 03:00:00 | 6.02865 | 21.198 |
| 2022-03-29 03:00:00 | 2022-04-13 03:00:00 | 5.51855 | 21.3446 |
| 2022-04-13 03:00:00 | 2022-04-28 03:00:00 | 5.62503 | 21.4268 |
| 2022-04-28 03:00:00 | 2022-05-13 03:00:00 | 6.28778 | 21.8067 |
| 2022-05-13 03:00:00 | 2022-05-28 03:00:00 | 7.93508 | 22.7945 |
| 2022-05-28 03:00:00 | 2022-06-12 03:00:00 | 6.529 | 22.3675 |
| 2022-06-12 03:00:00 | 2022-06-27 03:00:00 | 6.11689 | 22.4656 |
| 2022-06-27 03:00:00 | 2022-07-12 03:00:00 | 5.9794 | 22.1151 |

## Q4

Σκοπός του ερωτήματος αυτού είναι να βρεθούν οι τρεις μεγαλύτερες ώρες αιχμής ανά ημέρα της εβδομάδος, εννοώντας τις ώρες (π.χ., 7-8πμ, 3-4μμ, κλπ) της ημέρας με τον μεγαλύτερο αριθμό επιβατών σε μια κούρσα ταξί. Ο υπολογισμός αφορά όλους τους μήνες. Η υλοποίηση του πρέπει να γίνει με χρήση του DataFrame/ SQL API.

Το αποτέλεσμα του ερωτήματος είναι το εξής:

|  |  |  |  |
| --- | --- | --- | --- |
| hour\_of\_the\_day | week\_day | max\_passengers | rank |
| 19 | Friday | 9 | 1 |
| 18 | Friday | 9 | 2 |
| 3 | Friday | 8 | 3 |
| 1 | Monday | 9 | 1 |
| 17 | Monday | 9 | 2 |
| 20 | Monday | 9 | 3 |
| 13 | Saturday | 9 | 1 |
| 14 | Saturday | 9 | 2 |
| 6 | Saturday | 9 | 3 |
| 16 | Sunday | 9 | 1 |
| 21 | Sunday | 8 | 2 |
| 15 | Sunday | 8 | 3 |
| 21 | Thursday | 9 | 1 |
| 2 | Thursday | 9 | 2 |
| 12 | Thursday | 8 | 3 |
| 11 | Tuesday | 9 | 1 |
| 19 | Tuesday | 9 | 2 |
| 20 | Tuesday | 9 | 3 |
| 9 | Wednesday | 9 | 1 |
| 4 | Wednesday | 8 | 2 |
| 20 | Wednesday | 8 | 3 |

## Q5

Σκοπός του ερωτήματος αυτού είναι να βρεθούν οι κορυφαίες πέντε ημέρες ανά μήνα στις οποίες οι κούρσες είχαν το μεγαλύτερο ποσοστό σε tips. Η υλοποίηση του πρέπει να γίνει με χρήση του DataFrame/ SQL API.

Το αποτέλεσμα του ερωτήματος είναι το εξής:

|  |  |  |  |
| --- | --- | --- | --- |
| day\_of\_month | month | average\_tip\_percentage | rank |
| 29 | 1 | 0.215483 | 1 |
| 15 | 1 | 0.195323 | 2 |
| 22 | 1 | 0.193373 | 3 |
| 30 | 1 | 0.192807 | 4 |
| 21 | 1 | 0.192767 | 5 |
| 4 | 2 | 0.195576 | 1 |
| 5 | 2 | 0.195341 | 2 |
| 6 | 2 | 0.194006 | 3 |
| 10 | 2 | 0.193552 | 4 |
| 17 | 2 | 0.19291 | 5 |
| 9 | 3 | 0.195558 | 1 |
| 12 | 3 | 0.19392 | 2 |
| 30 | 3 | 0.193293 | 3 |
| 24 | 3 | 0.192785 | 4 |
| 10 | 3 | 0.192741 | 5 |
| 1 | 4 | 0.191378 | 1 |
| 7 | 4 | 0.191248 | 2 |
| 6 | 4 | 0.190912 | 3 |
| 27 | 4 | 0.19032 | 4 |
| 28 | 4 | 0.189369 | 5 |
| 12 | 5 | 0.192142 | 1 |
| 4 | 5 | 0.191388 | 2 |
| 11 | 5 | 0.190291 | 3 |
| 10 | 5 | 0.189719 | 4 |
| 6 | 5 | 0.189681 | 5 |
| 16 | 6 | 0.190439 | 1 |
| 8 | 6 | 0.189675 | 2 |
| 23 | 6 | 0.189216 | 3 |
| 9 | 6 | 0.189113 | 4 |
| 17 | 6 | 0.188299 | 5 |

# Χρόνοι εκτέλεσης

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API | Data frame/ SQL | | RDD | |
| # of workers | 1 | 2 | 1 | 2 |
| Q1 | 14.468 | 8.132 |  | |
| Q2 | 11.022 | 8.655 |
| Q3 | 6.309 | 8.155 |  |  |
| Q4 | 10.343 | 8.221 |  | |
| Q5 | 6.989 | 6.121 |

Οι παραπάνω χρόνοι εκτέλεσης είναι σε δευτερόλεπτα (seconds).