Project: Creation of a NoSQL Database for Credit Card Fraud Detection

Payment card fraud is a major challenge for business owners, payment card issuers, and transactional services companies, causing every year substantial and growing financial losses.

Many Machine Learning (ML) approaches have been proposed in the literature that tries to automate the process of identifying fraudulent patterns from large volumes of data. The book "Machine Learning for Credit Card Fraud detection – Practical handbook"[1] reports different approaches for facing the issue and for evaluating the quality of the proposed prediction results.

For the purpose of this project, we are not interested in the application and verification of ML approaches but we wish to exploit the transaction data simulator code (reported in Section 2 of Chapter 3 of the cited book https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_3_GettingStarted/SimulatedDataset.html) for the generation of data to be exploited for feeding a NoSQL database.

The simulator has the purpose to generate 3 tables:

1. **Customer profile**. Each customer has a unique identifier and is associated with the following properties: geographical location, spending frequency, and spending amounts. Moreover, the list of terminals on which the customer makes transactions is associated with his profile
2. **Terminal profile**. Terminal properties consist of a geographical location.
3. **Transactions**. This table reports for each transaction, the customer identifier, the terminal identifier, the amount that has been paid, and the date on which the transaction occurred. Some transactions can be marked as fraudulent.

Details on these tables and the Python scripts for the generation of the tables can be found at: fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_3_GettingStarted/SimulatedDataset.html

The following activities should be carried out:

1. You have to use the scripts for the generation of at least 3 datasets (each one containing the three tables described above) at an increased size (at least 50 Mbyte, 100 Mbyte, 200 Mbyte). Each dataset should have the same number of customers and terminals (at least 1000 customers and 200 terminals). You have to increase the number of transactions.
2. Define a conceptual model for the considered domain.
3. Choose one of the NOSQL systems considered in the course (MongoDB, Neo4J, PGQL in Oracle) and provide a data modeling to optimize the execution of the following operations:
    a. For each customer X, identify the customer Y (or the costumers) that share at least 3 terminals in which Y executes transactions and the spending amount of Y differs less than the 10% with respect to that of X. Return the name of X, the spending amount of X, the spending amount of the related costumer Y and the spending amount of Y.
    b. For each terminal identify the possible fraudulent transactions of the current month. The fraudulent transactions are those whose import is higher than 20% of the average import of the transactions executed on the same terminal in the previous month.
    c. Given a user u, determine the "co-customer-relationships CC of degree k". A user u' is a co-customer of u if you can determine a chain "u_1-t_1-u_2-t_2-…t_{k-1}-u_k" such that u_1=u, u_k=u', and for each 1<=l,j<=k, u_i <> u_j, and t_1,..t_{k-1} are the terminals on which a transaction has been executed. Therefore, $CC_k(u)$={u'| a chain exists between u and u' of degree k}. Please, note that depending on the adopted model, the computation of $CC_k(u)$ could be quite

[1] https://fraud-detection-handbook.github.io/fraud-detection-handbook

complicated. Consider therefore at least the computation of $CC_3(u)$ (i.e. the co-costumer relationships of degree 3).

d. Extend the logical model that you have stored in the NOSQL database by introducing the following information (pay attention that this operation should be done once the NOSQL database has been already loaded with the data extracted from the datasets):
    i. Each transaction should be extended with:
        1. The period of the day {morning, afternoon, evening, night} in which the transaction has been executed.
        2. The kind of products that have been bought through the transaction {high-tech, food, clothing, consumable, other}
        3. The feeling of security expressed by the user. This is an integer value between 1 and 5 expressed by the user when conclude the transaction.

    The values can be chosen randomly.
    ii. Customers that make more than three transactions from the same terminal expressing a similar average feeling of security should be connected as "buying_friends". Therefore also this kind of relationship should be explicitly stored in the NOSQL database and can be queried. Note, two average feelings of security are considered similar when their difference is lower than 1.
e. For each period of the day identifies the number of transactions that occurred in that period, and the average number of fraudulent transactions.

4. Create a script for loading the considered datasets in the chosen NOSQL and develop the scripts for implementing the previous operations. Take in mind that depending on the identified system, primitives of the query language should be embedded in a Python script.
5. Evaluate the execution times for all the previous operations (also the one for updating the NOSQL datastore) in the generated datasets.

3. What to deliver

The project can be delivered at any time of the academic year and is valid until April 2026. After this date, a new project will be assigned.
The project must be carried out by groups of at most 2 students.
You have to submit a zip file containing the following documents:

1. A single PDF file containing the technical documentation. The document must contain:
    a. The UML class diagram with explanations, motivations, constraints, and any other information that allows understanding the design carried out (it encompasses also the modification expressed on the initial data).
    b. The logical data model that has been realized for addressing the requirements imposed by the proposed operations. It is mandatory the presence of the motivations that led to the generation of the data model.
    c. The description of the script for loading the datasets in the chosen NOSQL system
    d. The description of the scripts for the implementation of the required operations
    e. A discussion of the performances obtained by the execution of the operations on the three considered datasets (included the loading operations). Please, discuss the eventual application of patterns for improving the performances of the considered operations. Please, include graphics for comparing the obtained performances.
2. All the scripts that have been developed and the information for reproducing the experiments that you have carried out.