Monash University

FIT5202 - Data processing for Big Data (S2 2024)

Assignment 1: Analysing Fraudulent Transaction Data

Due Date: 23:55 Friday 30/Aug/2024 (End of week 6)

Weight: 10% of the final marks

Background

In the dynamic world of eCommerce, the convenience and accessibility of online shopping have revolutionized consumer behavior and business operations. However, this digital transformation has also introduced significant challenges, particularly in the form of fraudulent transactions. Fraudulent transactions in eCommerce refer to unauthorized or deceitful activities conducted with the intent to steal money, goods, or sensitive information. These activities can take various forms, including credit card fraud, identity theft, account takeover, and phishing attacks. Fraudsters employ increasingly sophisticated techniques to exploit vulnerabilities in online systems, making it imperative for eCommerce platforms to implement robust detection mechanisms.

Fraud detection is essential for regulatory compliance and operational efficiency. Many regions have stringent regulations regarding data protection and financial transactions. Effective fraud detection ensures compliance with these regulations, avoiding legal repercussions and fines. Operational efficiency is another significant benefit of effective fraud detection. Fraudulent transactions often result in chargebacks, which can be costly and time-consuming to resolve. By detecting and preventing fraud, eCommerce platforms can reduce the frequency of chargebacks and streamline their operations. This not only saves time and resources but also allows businesses to focus on growth and innovation.

Big data processing plays a pivotal role in fraud detection by enabling the analysis of vast amounts historical data and streaming data in real-time, which is crucial for identifying and mitigating fraudulent activities. One of the primary advantages of big data in this context is its ability to perform real-time analytics. Traditional fraud detection methods often rely on retrospective analysis, which may not be timely enough to thwart sophisticated fraudsters. In contrast, big data analytics allows for the instant analysis of transactions, user behavior, and patterns, enabling organizations to identify and respond to potential fraud as it happens. Additionally, the integration of machine learning (ML) with big data analytics enhances predictive fraud prevention. These ML algorithms learn from historical data, identifying patterns and trends associated with fraudulent activities, and continuously evolve to adapt to new fraud tactics. This proactive defense mechanism helps in predicting and preventing fraud before it occurs, providing a robust layer of security for businesses and consumers alike.

In assignment 1, we will perform historical data analysis using Apache Spark. We will use RDD, DataFrame and SQL API learnt from week 1-4. Laster in assignment 2 (part A and B), we will explore using various ML techniques to build a fraud prediction model and streaming processing for realtime fraud detection.

The Dataset

The dataset can be downloaded from: https://bit.ly/3AbXtgU

You will find the following files after extracting the zip file:

1) transactions.csv: Contains transaction records.
2) merchant.csv: Contains basic information about merchants
3) customer.csv: Contains customer information.
4) category.csv: Contains product category information
5) geolocation.csv: Contains geographic location information.

Metadata of the dataset can be found in the appendix at the end of this document.

Assignment Information

The assignment consists of three parts: Working with **RDD**, Working with **Dataframes**, and Comparison of three forms of Spark abstractions. In this assignment, you are required to implement various solutions based on RDDs and DataFrames in PySpark for the given queries related to eCommerce data analysis.

Getting Started

- Download your own dataset from the URL provided in Moodle.
- Download a template file for submission purposes:
    - ***A1_template.ipynb*** file in Jupyter notebook to write your solution. Rename it into the format (for example: ***A1_xxx0000.ipynb*****.** This file contains your code solution(xxx0000 is your authcode).
- You will be using Python 3+ and PySpark 3.5.0 for this assignment (The environment is provided as a Docker image, the same you use in labs.)

Part 1: Working with RDDs (30%)

In this section, you need to create RDDs from the given datasets, perform partitioning in these RDDs and use various RDD operations to answer the queries.

1.1 Data Preparation and Loading (5%)

1. Write the code to create a SparkContext object using SparkSession. To create a SparkSession you first need to build a SparkConf object that contains information about your application, use **Melbourne time** as the session timezone. Give an appropriate name for your application and run Spark locally with **4 cores on your machine**.
2. Load csv files into multiple RDDs.

3. For **each** RDD, remove the header rows and display the total count and first 10 records. (Hint: You can use csv.reader to parse rows into RDDs.)
4. Drop personal information columns from RDDs: cc_num, firstname, lastname, address.

1.2 Data Partitioning in RDD (15%)

1. For **each** RDD, print out the total number of partitions and the number of records in each partition (5%).
2. Answer the following questions:
    a. How many partitions do the above RDDs have?
    b. How is the data in these RDDs partitioned by default, when we do not explicitly specify any partitioning strategy? Can you explain why it is partitioned in this number?
    c. Assuming we are querying the dataset based on transaction date, can you think of a better strategy to partition the data based on your available hardware resources?

    Write your explanation in Markdown cells. (5%)
3. Create a user defined function (UDF) to transform trans_timestamp to ISO format(YYYY-MM-DD hh:mm:ss), then call the UDF and add a new column **trans_datetime** (5%)

1.3 Query/Analysis (10%)

For this part, write relevant **RDD operations** to answer the following questions.

1. Calculate the summary of fraudulent transactions amount for each year, each month. Print the results in tabular format. (5%)
2. List 20 mechants that suffered the most from fraudulent activities (i.e. 20 highest amount of monetary loss). (5%)

Part 2. Working with DataFrames (45%)

In this section, you need to load the given datasets into PySpark DataFrames and use *DataFrame functions* to answer the queries.

2.1 Data Preparation and Loading (5%)

1. Load the CSV files into separate dataframes. When you create your dataframes, please refer to the metadata file and think about the appropriate data type for each column.
2. Display the schema of the dataframes.

Think about: When the dataset is large, do you need all columns? How to optimize memory usage? Do you need a customized data partitioning strategy? (**Note: You don't need to answer these questions.**)

2.2 Query/Analysis (40%)

Implement the following queries using **dataframes**. You need to be able to perform operations like filtering, sorting, joining and group by using the functions provided by the DataFrame API.

1. Transform the “trans_timestamp” to multiple columns: trans_year, trans_month, trans_day, trans_hour (24-hour format). (note: you can reuse your UDF from part 1 or create a new one.) (5%)
2. Calculate the total amount of fraudulent transactions for each hour. Show the result in a table and plot a bar chart. (5%)
3. Print number of small transactions(<=$100) from female who was born after 1990. (5%)
4. We consider a fraud-to-sales (F2S) ratio of 3% as a benchmark. If a merchant has F2S >= 3%, it is considered operating at very high rick. How many companies are operating at very high risk? (note: The answer should be a single number.) (5%)
5. “Abbott and Adam Group” wants to know their total revenue(sum of **non-fraud** amt) in **each state** they operate, show the **top 20 results by revenue in descending order**. You output should include merchant name, state and total revenue. (note: Abbott and Adam group include all merchants who name start with “Abbott” or “Adam”.) (10%)
6. For each year (2020-2022), aggregate the number(count) of fraudulent transactions every hour. Plot an appropriate figure and observe the trend. Write your observations from your plot (e.g. Is fraudulent activities increasing or decreasing? Are those frauds more active after midnight or during business hours?). (10%)

Part3: RDDs vs DataFrame vs Spark SQL (25%)

Implement the following queries using RDDs, DataFrame in SparkSQL **separately**. Log the time taken for each query in each approach using the “%%time” built-in magic command in Jupyter Notebook and **discuss the performance difference between these 3 approaches**.

We consider city with population < 50K as small(denoted as S); 50K-200K as medium(M), >200K as large(L). For each city type, using customer age bucket of 10(e.g. 0-9, 10-19, 20-29…), show the percentage ratio of fraudulent transactions in each age bucket.

(note: You can reuse the loaded data/variables from part 1&2.)

(hint: You may need to create intermediate RDD/dataframes for this query.)

1) Implement the above query using RDDs, DataFrame and SQL separately

and print the results. (note: The three different approaches should have the same results). (15%)

2) **Which one is the easiest to implement in your opinion? Log the time taken for each query, and observe the query execution time, among RDD, DataFrame, SparkSQL, which is the fastest and why? Please include proper reference. (Maximum 500 words.) (10%)**

Submission

You should submit your final version of the assignment solution online via Moodle. You must submit the files created:

- Your jupyter notebook file **(e.g., A1_authcate.ipynb**).
- **A pdf file** saved from jupyter notebook with all output following the file naming format as follows: **A1_authcate.pdf**

Note that the both submitted (jupyter and pdf) files will be scanned using plagiarism detection software. The highest similarity score among students may be interviewed to prove the originality of the work.

Assignment Marking Rubric

For each task individually, you'll be marked based on the quality of your work in a 3-level scale (0%, 50% and 100%).

- 0%: No attempt or incorrect answer with poor attempt;
- 50%: Partial mark for a good attempt but incorrect result;
- 100%: Full mark for correct attempt.

In your submission, the jupyter notebook file should contain the **code and its output**. It should follow *programming standards, readability of the code, organization of code*. Please find the PEP 8 -- Style Guide for Python Code for your reference. Here is the link: https://peps.python.org/pep-0008/ **Penalty up to 10% applies if your code is hard to understand with insufficient comments.**

Late submissions

Late Assignments or extensions will not be accepted unless you submit a special consideration form. ALL Special Consideration, including within the semester, is now to be submitted centrally. This means that students MUST submit an online Special Consideration form via Monash Connect. For more details, please refer to the **Unit Information** section in Moodle.

There is a **5% penalty per day including weekends** for a late submission. Also, the cut-off date is 7 days after the due date. No submission will be accepted (i.e. 0 mark) after the cut-off date unless you have a special consideration.

Mark Release and Review

- Mark will be released within 10 business days after the submission deadline.
- Reviews and disputes regarding the mark will be accepted maximum 7 days after the release date (including weekend).

Other Information

Where to get help

You can ask questions about the assignment in the Assignments section in the Ed Forum accessible on the unit's Moodle Forum page. This is the preferred venue for assignment clarification-type questions. You should check this forum regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification. Also, you can attend scheduled consultation sessions if the problem and the confusions are still not solved.

Plagiarism and collusion

Plagiarism and collusion are serious academic offenses at Monash University. Students must not share their work with any other students. Students should consult the policy linked below for more information.
https://www.monash.edu/students/academic/policies/academic-integrity
See also the video linked on the Moodle page under the Assignment block.
Students involved in collusion or plagiarism will be subject to disciplinary penalties, which can include:

- The work not being assessed
- A zero grade for the unit
- Suspension from the University
- Exclusion from the University

Generative AI Statement

As per the University's policy on the guidelines and practice pertaining to the usage of Generative AI:

AI & Generative AI tools may be used SELECTIVELY within this assessment.

Where used, AI must be used responsibly, clearly documented and appropriately acknowledged (see Learn HQ).

Any work submitted for a mark must:

1) Represent a sincere demonstration of your human efforts, skills and subject knowledge that you will be accountable for.
2) Adhere to the guidelines for AI use set for the assessment task.
3) Reflect the University's commitment to academic integrity and ethical behaviour.

Inappropriate AI use and/or AI use without acknowledgement will be considered a breach of academic integrity.

The teaching team encourage students apply their own critical thinking and reasoning skills when working on the assessments with assistant from GenAI. Generative AI tools may produce inaccurate content and this could have a negative impact on students' comprehension of big data topics.

Data source acknowledgement:

The dataset is a remix based on several real-world dataset. All name, age, dob, salary etc. are randomly generated synthetic dataset.

Appendix: Metadata of the Dataset Schema

category.csv	
id_category	Unique identifier of category, Int4(32 bit integer)
category	Category name
customer.csv	
id_customer	Unique identifier of customer(String)
cc_num	Default credit card number for payment
firstname	Firstname
lastname	Lastname
gender	Gender
address	Address
job	Job Description
dob	Date of Birth
acct_num	Account Number
id_geolocation	(FK) Geolocation ID
geolocation.csv	
id_geolocation	Unique identifier of Geo Location
city	City
state	State
zip	Postcode(Zip Code)
lat	Latitude
long	Longititude
population	City Population
Note: 1) Different state may have same city name; 2) The same city in same state may have multiple postcode, the population figures are for the whole city.	
merchant.csv	
id_merchant	Unique Identifier of Merchant
id_geolocation	(FK) Geo Location of a merchant

merchant	Merchant Name (note: A merchant may have many stores at different geo locations.)
transactions.csv	
id_transaction	Unique Identifier of a transaction
trans_timestamp	Transaction Timestamp(Unix Timestamp)
amt	Transaction Amount
is_fraud	If the transaction is fraud or not. (1=True, 0=False)
id_customer	(FK) Customer ID
id_geolocation	(FK) Geolocation ID
id_merchant	(FK) Merchant ID
id_category	(FK) Category ID