Monash University

FIT5202 - Data processing for Big Data 2025 S1

Assignment 2B: Using Real-time Streaming Data and Machine Learning to Predict and Visualise eCommerce Fraud

Due: **(Week 12, Friday 23:55 30/May/2025)**
Worth: **15%** of the final marks

Background

In the dynamic world of eCommerce, **M**onash **F**ashion **C**orporation (**MFC**, an imaginary company) has established itself as a leading online retailer of fashion products, catering to a diverse and global customer base. With a wide range of products, from trendy apparel to stylish accessories, MFC has built a reputation for quality and customer satisfaction. Our platform leverages advanced technologies to provide a seamless shopping experience, ensuring customers can easily browse, select, and purchase their desired items. However, as our business has grown, so too have the challenges associated with maintaining the integrity and security of our transactions. Unfortunately, the rise of digital commerce has been accompanied by increased fraudulent activities, which pose significant risks to our financial stability and customers' trust. To address these challenges, we are committed to implementing cutting-edge solutions to detect and prevent fraudulent transactions in real-time.

Problem Statement & Project Objective (See A2 Part A)

We have already developed the machine learning models in part A of the assignment. **In part B, we will create a prototype streaming application to demonstrate the integration of machine learning models, Kafka, and Spark streaming. First, we predict potential fraud in** real time as customers browse the website. Then, we visualise the data to help the company make better business decisions.

Based on customers' browsing behaviours, we will focus on two aspects:

1) Predicting the potential fraudulent transaction to warn the company's operation team.
2) Predicting potential inventory requirements can help the company optimise its inventory and logistics.

<u>Required Datasets in Moodle:</u>

- All files from assignment 2 Part A(A2A).
- Your saved best model from A2A
- new_browsing_behaviour.csv: (New browsing behaviour data to simulate real-time streaming)
- new_transaction.csv: (New transactions linked to browsing behaviour)

What you need to achieve

The company requests a prototype application to ingest the new browsing behaviour data and predict potential fraud. To achieve this, you need to simulate the streaming data production using Kafka and then build a streaming application that ingests the data and integrates the machine learning model to predict potential fraud.

Architecture

The overall architecture of the assignment setup is represented by the following figure.

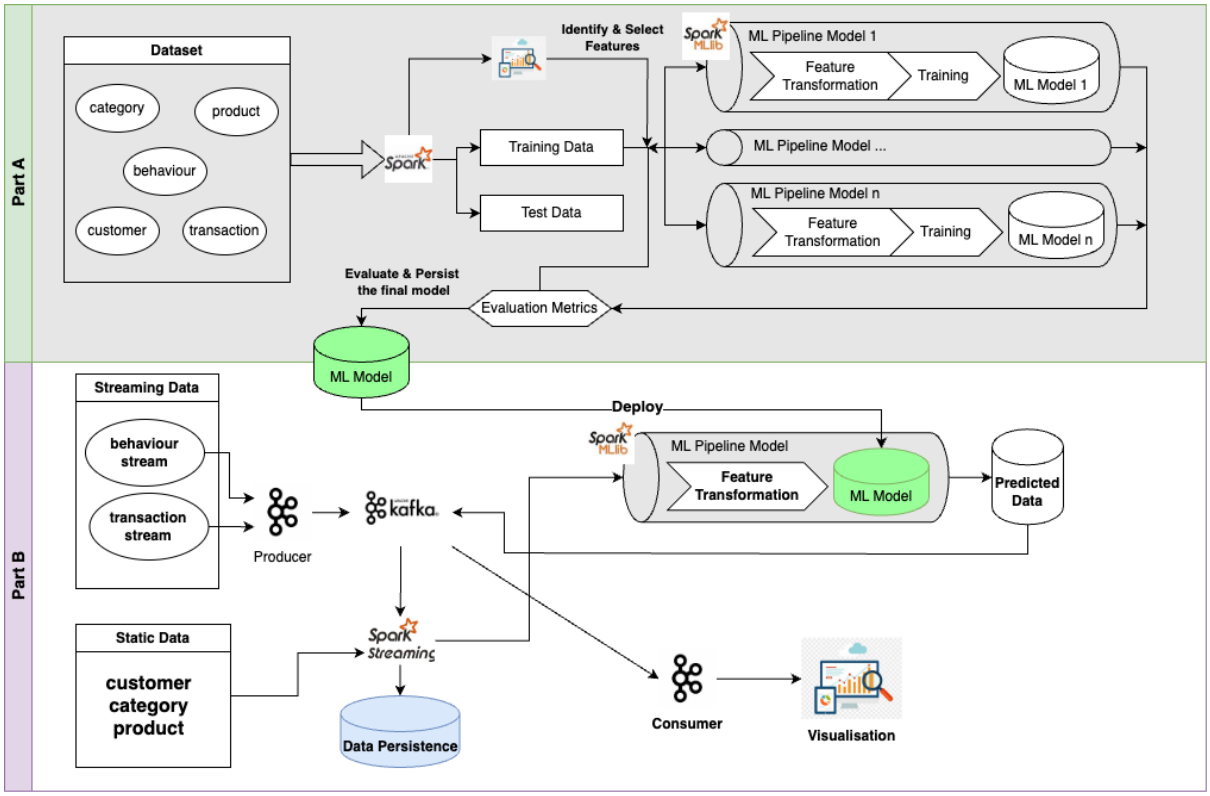

Fig 1: Overall Architecture for Assignment 2 (This assignment is Part B)

Part B of the assignment consists of creating data streams in Kafka and joining static data frames with streaming data to create features, using Spark streaming/MLLib to predict potential fraud/inventory requirements, and visualising predicted data.

Overview:

1. In task 1, you will read and publish the data to the Kafka stream.
2. In task 2, you will process the data streams using Spark Structured Streaming and PySpark ML/DataFrame.
3. For task 3, you will read the data from the Kafka stream from part 1 and visualise it.

Please follow the steps to document the processes and write the codes in the Jupyter Notebook.

Getting Started

- Download the dataset from Moodle.
- Download three ipynb template files from Moodle

- ***A2B-Task1_producer.ipynb*** file for streaming data production
- ***A2B-Task2_spark_streaming.ipynb*** file for consuming and processing data using Spark Structured Streaming
- ***A2B-Task3_consumer.ipynb*** file for consuming the data using Kafka and visualise

Part 1. Producing the data (15%)

In this task, we will implement Apache Kafka producers to simulate real-time data streaming. Spark is not allowed in this part since it's simulating a streaming data source.

1. Your program should send browsing behaviour data batch-by-batch to the Kafka stream. One batch consists of a **200 rows** from the browsing behaviour dataset. The CSV shouldn't be loaded to memory at once to conserve memory (i.e. Read row as needed).
    - Keep track of the **start and end** event_time of browsing behaviour data batch (You can assume the dataset is sorted by event_time.)

2. For each row, add a timestamp column named '**ts**' in Unix timestamp format. This is the event time when you send the streaming data, which should be in *int* format. The '**ts**' should be the same for the data sent in one batch.

3. Read the transaction data rows that fall between the start and end event_time of the browsing behaviour data recorded in task 1.1, and create a batch. Similarly, as in task 1.2, add the column named '**ts**' for each row.

4. At an interval of **1 second,** send the two data batches consecutively (browsing behaviour data & transaction data) from 1.1 and 1.3 to respective Kafka topics with an appropriate name (i.e., wait for **1 second** before sending the next two data batches). Fig. 1 depicts a sequence of data batches sent to Kafka and the data structures.

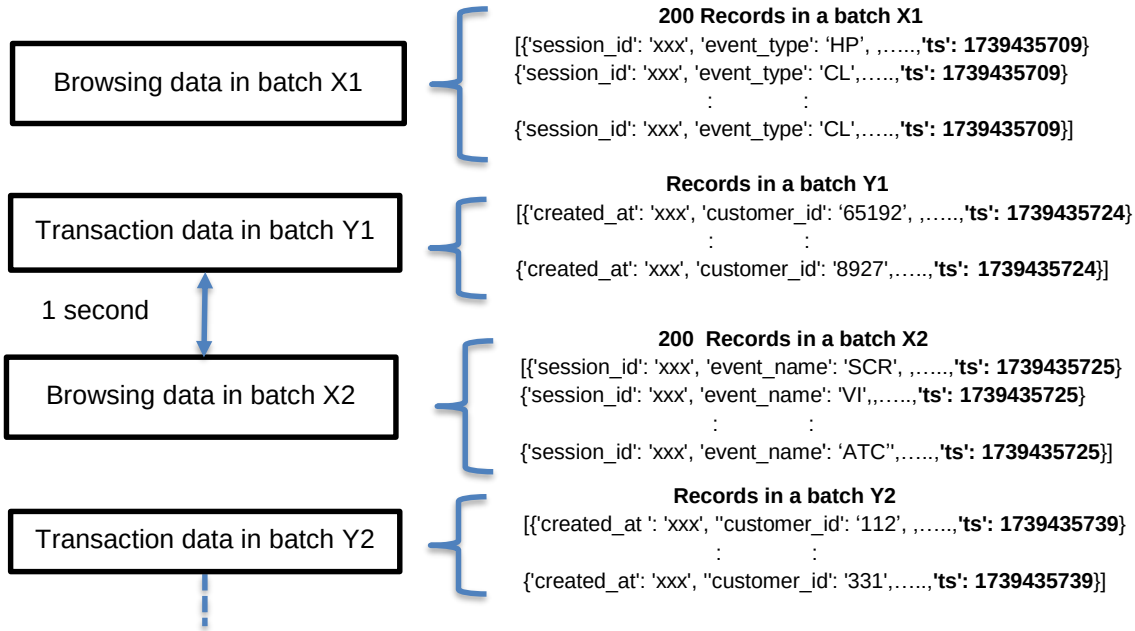

Fig. 1. Illustration of data batches to be sent.

Note 1: All the data except for the 'ts' column should be sent in the original String type without changing to any other type. This is because we are simulating a streaming access log and need to reduce the required processing at the source.

Save your code in ***Assignment-2B-Task1_producer.ipynb***.

Part 2. Streaming application using Spark Structured Streaming (70%)

In this task, we will implement Spark Structured Streaming to consume the data from task 1 and perform predictive analytics.

Important:

- **This task uses PySpark Structured Streaming with PySpark Dataframe APIs and PySpark ML.**
- **You also need your pipeline model from A2A to make predictions and persist the results.**
- **If you didn't complete A2A or failed to build an ML model, we still encourage you to attempt this part. You can use the "is_fraud" label to complete many tasks in this part; however, the maximum mark you will receive is capped at 80% (below HD).**

1. Write code to create a SparkSession, which 1) uses **four cores** with **a proper application name**; 2) use the Melbourne timezone; 3) ensure a checkpoint location has been set.

2. Similar to assignment 2A, write code to define the data schema for the data files, following the data types suggested in the metadata file. Load the static datasets (e.g. customer, product, category) into data frames. (You can use your code from 2A.)

3. Using the Kafka topics from the producer in Task 1, ingest the streaming data (i.e., the browsing behaviour and transaction data) into Spark Streaming, assuming all data comes in the **String** format. Except for the 'ts' column, you shall receive it as an **Int** type.

4. Then, the streaming data format should be transformed into the proper formats following the metadata file schema, similarly to assignment 2A. Perform the following tasks:
    a) For the 'ts' column, convert it to the timestamp format, we will use it as **event_ts**.
    b) If the data is late for more than 2 minutes, discard it. (Hint: Use watermaking)

5. Aggregate the streaming data frames and create features you used in your assignment 2A model. (note: customer ID has already been included in the stream.) Then, join the **static** data frames with the streaming dataframes as our final data for prediction. Perform data type/column conversion according to your ML model and print out the Schema. (Again, you can reuse code from A2A). (Hints: For stream-stream join, you need to apply watermarking to one/both streams based on event time)

6. The company is interested in the number of potential frauds as they happen and the products in customers' shopping carts (so that they can plan their stock level ahead.). Load your ML model, and use the model to predict if there is a fraud or not in each browsing session/transaction. Persist the prediction result in parquet format, then read the parquet result and show the results. (You can stop the parquet streaming after you showed the results)

7. Using the prediction results, write code to process the data following the requirements below and show results. All results must be shown in the Jupiter file.
    a) Every 10 seconds, show the total number of frauds (prediction = 1) in the **last 2 minutes.** (Hint: Use window size of 2 minutes with 10-second slide).
        - Show results every 10 seconds (Use trigger interval of 10 seconds). For example, it should be like the following screenshot. (You can stop this stream after some results were showed). Hint: Use foreachBatch as output sink.

```
+------------------------------------------+-----------+
|window                                    |fraud_count|
+------------------------------------------+-----------+
|{2025-02-13 13:51:00, 2025-02-13 13:53:00}|1          |
|{2025-02-13 13:51:10, 2025-02-13 13:53:10}|1          |
+------------------------------------------+-----------+
```

    b) Every 30 seconds, show the total quantity of products for non-fraud transactions (i.e., when prediction=0). You can assess to this information by extracting from the customer shopping cart detail from product metadata in the transaction data.

    c) Every 1 minute, find the top 20 products by total quantity. Show their product ID, name and total quantity. Again, we only need the non-fraud transactions (prediction=0) by extracting customer shopping cart details.

Save your code in ***Assignment-2B-Task2_spark_streaming*.ipynb**.

Part 3. Consuming data using Kafka and Visualise (15%)

In this task, we will implement an Apache Kafka consumer to consume the data from Part 1.

Important:

- **In this part, Kafka consumers are used to consume the streaming data published from Part 1.**
- **Please do not use Spark in this part. It's OK to use Pandas or any Python library to do simple calculations for the visualisation.**

1. Write a python program that consumes the streaming transaction data sent by the producer in Part 1 using kafka consumer.

2. Your program should also get the number of successful transactions (i.e., when 'payment_status' is 'Success') for each batch of transaction data, and visualise these counts in real time based on the event timestamp.
    a) Use a line chart to visualize the counts of successful transactions over event timestamp.
    b) Use ‘ts’ as timestamp generated from producer step. Hint - x-axis can be used to represent the timestamp, while y-axis can be used to represent the counts of successful transactions.
    c) Note: this plot shall be real-time plot, which will be updated if new streaming data comes in from the Producer in Part 1.

Save your code in ***Assignment-2B-Task3_consumer.ipynb***.

Assignment Marking Rubric

Detailed mark allocation is available in each task. For complex tasks and explanation questions, you will receive marks based on the quality of your work.

In your submission, the jupyter notebook file should contain the **code and its output**. It should follow *programming standards, readability of the code, and organisation 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 applies if your code is hard to understand with insufficient comments.

Submission

You should submit your final version of the assignment solution via Moodle. You must submit the following:

- A **zip** file named based on your authcate name (e.g. abcd1234). The zip file should contain
    - ***Assignment-2B-Task1_producer_authcate*.ipynb**
    - ***Assignment-2B-Task2_spark_streaming_authcate*.ipynb**
    - ***Assignment-2B-Task3_consumer_authcate.ipynb***

  The file in submission should be a ZIP file and *not any other kind of compressed folder (e.g. .rar, .7zip, .tar).* Please do not include the data files in the ZIP file.
- The assignment submission should be uploaded and finalised by **(Week 12, Friday 23:55 30/May/2025 (Malaysia Time)).**

Other Information

Where to get help

You can ask questions about the assignment in the Assignments section in the Ed Forum, which is 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 confusion are still unresolved.

Searching and learning on commercial websites/forums (e.g. Quora, Stack Overflow) is allowed. However, you should not post/ask assignment questions on those forums.

Plagiarism and collusion

Plagiarism and collusion are severe academic offences 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

Late submissions and Special Consideration

ALL Special Consideration, including within the semester, is now handled 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. zero 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 a maximum of 7 days after the release date (including weekends).

Generative AI Statement

As per the University's policy on the guidelines and practices 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 to 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 datasets. Transaction records are from real-world data, user name, age, dob, salary, etc. are randomly generated synthetic datasets.

We thank the authors/owners for sharing the original datasets.

1. eCommerce behavior data from multi category store | Kaggle
2. REES46
3. E-Commerce Data | Kaggle
4. Brazilian E-Commerce Public Dataset by Olist | Kaggle
5. Geoscape Geocoded National Address File (G-NAF) - Dataset - data.gov.au
6. Popular Baby Names - Dataset - data.sa.gov.au
7. Fashion Campus | Kaggle