Monash University

FIT5202 - Data processing for Big Data 2025 SSB

Assignment 2: Using Streaming and ML for Food Delivery Prediction and Visualisation

Weight: 30% (15% A2A+15% A2B)

Background

Food delivery services have become an integral part of modern society, revolutionising the way we consume meals and interact with the food industry. These platforms, accessible through websites and mobile apps, provide a convenient bridge between restaurants and consumers, allowing users to browse menus, place orders, and have food delivered directly to their doorstep with just a few taps. In today's fast-paced world, where time is a precious commodity, food delivery services offer an invaluable solution, catering to busy lifestyles, limited mobility, and the ever-present desire for convenience.

In the food delivery industry, accurate on-time delivery prediction is paramount. Big data processing allows companies to achieve this by analysing vast datasets encompassing order details, driver performance, real-time traffic, and even weather.

Objective of the Project

In the **first assignment (A1)**, we performed data analysis with the datasets to uncover key trends and patterns related to delivery times, order volumes, and other crucial metrics.

In **assignment 2A**, we will harness the power of Apache Spark's MLLib to construct and train machine learning models. We will focus on accurately and efficiently predicting delivery times.

Finally, assignment **2B** will utilise Apache Spark Structured Streaming and your ML model from 2A to process live data streams and dynamically make predictions.

Key Information

This is a two-part assignment (A2A and A2B) that requires staged submissions. In part A2A, you are going to use the provided dataset, complete the assignment tasks, and build your ML model; then, in part A2B, the trained ML model will be used in combination with streaming data to make real-time predictions.

A2A Due Date: **(23:55 Friday 31/Jan/2025, End of Week 5)**

A2B Due Date: **(23:55 Wednesday 5/Feb/2025, Mid of Week 6)**

Submission links can be found in Moodle.

Weight: 30% of Final Marks (15% each for 2A and 2B) A2A and A2B will be marked separately.

A2B has a compulsory interview/demo component, which will be conducted during the last lab. The teaching team only marks A2B submissions during your demo session. Failure to attend this demo will result in 0 marks (for A2B).
(Please pay attention to the unit announcement in the final teaching week. If you have an extension/special consideration, more demo sessions will be arranged after the exam.)

The Datasets:

- **driver.csv (From A2A)**
- **delivery_address.csv (From A2A)**
- **restaurant.csv (From A2A)**
- **new_order.csv (for A2B)**

What you need to achieve

Use Case 1 (A2A)	Based on the historical dataset, build a ML model that can predict food delivery time.	Regression
Use Case 2 (A2B)	Use streaming data to perform real-time prediction and visualise the results	Spark Structured Streaming

Architecture

The following figure represents the overall architecture of the assignment setup.

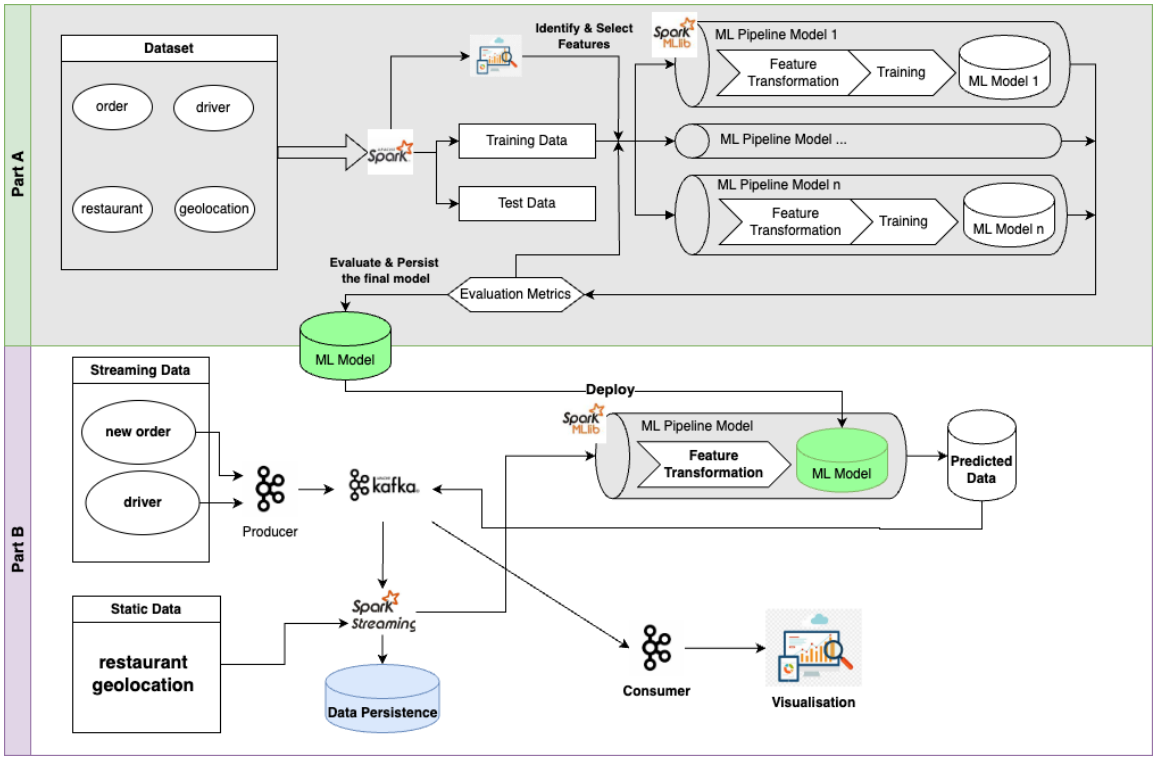

Fig 1: Overall Architecture for Assignment 2

In both parts, you must implement the solutions using PySpark DataFrame/MLlib for the data pre-processing and machine learning pipelines. Excessive use of Pandas for data processing is discouraged. Please follow the steps to document the processes and write the codes in your Jupyter Notebook.
In **Part B,** you will utilise the model from part A and generate operational insights for the food delivery platform.

Getting Started

- Download the datasets from Moodle.
- Download a template file for submission purposes:
    - ***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
    - For submission, please append your authcate ID at the end of the filename. ***(e.g. A2B-Task1_producer_xxxx0000.ipynb*** **xxxx0000** is your authcate ID.
- You will use Python 3+ and PySpark 3.5.0+ for this assignment (This environment is the same as we used in labs.)

IMPORTANT:
Please answer each question in your Jupyter Notebook file using code/markdown cells. Acknowledge any ideas or codes you referenced from others in the markdown cell or reference list.
If you use generative AI tools, all prompts you use should also be included in the reference section or appendix.

A2 Part B Specification

Part 1. Producing the data (10%)

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

1. Your program should send one batch of order data **every 5 seconds**. One batch consists of a **random 20-50 rows (i.e. orders)** from the new_order.csv. The CSV shouldn't be loaded to memory at once to conserve memory (i.e. Read row as needed). (note: you can read the file sequentially; only the number of rows is random.)
2. Update **order_ts** to the current timestamp and spread your batch out evenly for 5 seconds. Update **ready_ts** if necessary (when you use it in your ML model).
    a. For example, assume you send a batch of 50 records at 2025-01-26 00:00:00 (ISO format: YYYY-MM-DD HH:MM:SS) -> (ts = 1737810000):
        - Record 1-10: ts = 1737810000
        - Record 11-20: ts = 1737810001
        - Record 21-30: ts = 1737810002
        - ….
3. Set the **delivery_persion_id and delivery_time** to null (because the order just came in, we haven't assigned a delivery person yet).
4. Send your order batch to a Kafka topic with an appropriate name.
5. Every 5 seconds, randomly select **50** drivers from the driver dataset, send it to another Kafka topic. (note: this simplifies the application, ensuring all orders have at least one available driver.)

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

Part 2. Streaming application using Spark Structured Streaming (50%)

In this task, you will implement Spark Structured Streaming to consume the data from task 1 and perform prediction.

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.**

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. 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. restaurants, delivery_address) into data frames. (You can reuse your code from 2A.)
3. Using the Kafka topic (orders) from the producer in Task 1, ingest the streaming data into Spark Streaming, assuming all data comes in the **String** format. Except for the 'order_ts' column, you shall receive it as an **Int** type.

4. Then, the streaming data frames (orders and drivers) should be transformed into the proper formats following the metadata file schema, similar to assignment 2A.
5. From each order, a) select **5** random drivers; b) use your ML model to predict their delivery time; c) select the fastest driver (i.e. the shortest delivery time), assign the driver to the order and then update delivery_time with your prediction.
    - Note 1: You may need to join other data frames like restaurant and delivery_address if you used them in your model.
    - Note 2: Assume one driver can only carry one order at a time within a batch. Your "random 5" selection should exclude those drivers who have already been assigned to an order.
6. Perform the following aggregations:
    a) Every 15 seconds, show the total number of revenue (sum of order_total) for each type of order (drinks, meals, snacks, etc.).
    b) Every 30 seconds, for each suburb of restaurants, count the number of orders with predicted delivery time <=15 minutes and > 15 minutes.
7. Save the data from 6a and 6b to a Parquet file as streams. (Hint: Parquet files support streaming writing/reading. The file keeps updating while new batches arrive.)
8. Read the two parquet files from task 7 as **data streams** and send them to Kafka topics with appropriate names.
   (Note: **You shall read the parquet files as a streaming data frame and send messages to the Kafka topic when new data appears in the parquet file.)**

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

Part 3. Consuming data using Kafka and Visualise (20%)

In this task, we will implement an Apache Kafka consumer to consume the data from Part 2.

Important:

- **In this part, Kafka consumers are used to consume the streaming data published from task 2.8.**
- **This visualisation part doesn't require parallel processing, so please do not use Spark. It's OK to use Pandas or any Python library to do simple calculations for the visualisation.**

1. (**Basic plot**) Plot a diagram to show data from 6a (i.e. every 15 seconds, plot the total number of revenues for each type of order.) You are free to choose the type of plot.
2. (**Advanced plot**) Plot a choropleth or bubble map to visualise data from 6b (restaurant's suburb-based order count for <=15 and >15 minutes; you may use different colors or subplots.).
   Choropleth: https://python-graph-gallery.com/choropleth-map/
   Bubble Map: https://python-graph-gallery.com/bubble-map/

Note: Both plots shall be real-time plots, which will be updated if new streaming data comes in from part 2. For the advanced plot, if you need additional data for the plots, you can add them in part 2.

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

Part 4: Demo and Interview (20%)

IMPORTANT: The interview is compulsory, and we only mark your A2B during the interview. No marks will be awarded if the interview is not attended. (0 marks for the whole A2B, not just this section).

The demo/interview session details will be announced/arranged in the last teaching week. Please pay attention to the unit announcement email and Ed forum.

Each demo is roughly 10 minutes. You have 5-6 minutes to show your application; then, your marker will ask 3-4 questions to assess your understanding. Please come to your allocated demo session a few minutes early and ensure your laptop/application works correctly.

Demo/Interview is marked on a 5-level scale:

The demo is working, and the student has a competent understanding	20
Working demo, partial understanding	15
The demo is not working, partial understanding	10
The demo is not working, low understanding	5
No attendance or can't answer most of the questions	0

For those students with special consideration and extension, more demo sessions will be arranged after the exam.

Submission A2B

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 A2B due date is **23:55 Wednesday 5/Feb/2025**

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.

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 serious academic offences at Monash University. Students must not share their work with 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 for which you will be accountable.
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 encourages students to apply their own critical thinking and reasoning skills when working on the assessments with assistance from GenAI. Generative AI tools may produce inaccurate content, which could have a negative impact on students' comprehension of big data topics.

Appendix: Metadata of the Dataset Schema

(note: Some self-explanatory columns are not included. i.e. street_name, postcode)

delivery_address.csv	
gid	ID of deliver address geolocation
latitude	Latitude, Decimal(10,8)
longitude	Longitude, Decimal(11,8)
geom	Geometry point on maps
delivery_id	ID of a delivery address, this ID shall be used in join queries.
driver.csv	
driver_id	Unique identifier of delivery person/driver
age	Delivery driver's age (Integer), range is 18-60.
rating	Overall rating of the driver (float, 0-5 scale)
year_experience	Years of delivery experience, which may affect delivery speed
vehicle_condition	A driver's vehicle condition (from 0 – Excellent, 1 - Good, 2 - Fair, 3 - Poor)
type_of_vehicle	Motorcycle, Scooter, electric_scooter, etc. (String)
order.csv	
order_id	Unique identifier of an order
delivery_person_id	Unique ID of the driver delivering an order
order_ts	timestamp when an order is placed
ready_ts	timestamp when a restaurant finishes preparing an order, i.e. ready for the delivery driver to pick up.
weather_condition	Weather condition at the time of order (Sunny, Windy, Storm, etc.) (String)
road_condition	Road traffic conditions (Low, Medium, Jam, etc.)
type_of_order	Snacks, Meal, Drinks, etc. (String)
order_total	Total value of the order.
delivery_time	Time taken for the delivery, not including preparation time. This column can be used as our label column.

Travel_distance	Total travel distance for the delivery.
restaurant_id	ID of a restaurant.
delivery_id	ID of a delivery address.
restaurants.csv	
row_id	Row id of the restaurant in database (not used as a dataset).
Restaurant_code	Internal code of a restaurant
Chain_id	If a restaurant belongs to a chain (empty if not).
Primary_cuisine	Primary Cuisine of the restaurant
geom	Geometry point of the restaurant
Restaurant_id	ID of a restaurant (used for join as primary key).